Local features for view matching across independently moving cameras

by

Alessio Xompero

Bachelor in Electronics and Telecommunications Engineering 2012

Master in Telecommunications Engineering 2015

A dissertation submitted to

The School of Electronic Engineering and Computer Science

in partial fulfilment of the requirements of the Degree of

Doctor of Philosophy

in the subject of

Electronic Engineering

Queen Mary University of London

Mile End Road

E1 4NS, London, UK

September, 2019

Declaration

I, Alessio Xompero, confirm that the research included in this thesis is my own work, that is duly acknowledged, and my contributions are indicated. I have also acknowledged previously published materials.

I attest that reasonable care has been exercised to ensure the originality of this work, and, to the best of my knowledge, does not break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the college has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree to any other university.

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Signature: Alessio Xompero

Date: September 24, 2019

Primary supervisor	Secondary supervisor	Author
Prof. Andrea Cavallaro	**Dr. Oswald Lanz**	**Alessio Xompero**

Local features for view matching across independently moving cameras

Abstract

Moving platforms, such as wearable and robotic cameras, need to recognise the same place observed from different viewpoints in order to collaboratively reconstruct a 3D scene and to support augmented reality or autonomous navigation. However, matching views is challenging for independently moving cameras that directly interact with each other due to severe geometric and photometric differences, such as viewpoint, scale, and illumination changes, can considerably decrease the matching performance. This thesis proposes novel, compact, local features that can cope with with scale and viewpoint variations. We extract and describe an image patch at different scales of an image pyramid by comparing intensity values between learnt pixel pairs (binary test), and employ a cross-scale distance when matching these features. We capture, at multiple scales, the temporal changes of a 3D point, as observed in the image sequence of a camera, by tracking local binary descriptors. After validating the feature-point trajectories through 3D reconstruction, we reduce, for each scale, the sequence of binary features to a compact, fixed-length descriptor that identifies the most frequent and the most stable binary tests over time. We then propose XC-PR, a cross-camera place recognition approach that stores locally, for each uncalibrated camera, spatio-temporal descriptors, extracted at a single scale, in a tree that is selectively updated, as the camera moves. Cameras exchange descriptors selected from previous frames within an adaptive temporal window and with the highest number of local features corresponding to the descriptors. The other camera locally searches and matches the received descriptors to identify and geometrically validate a previously seen place. Experiments on different scenarios show the improved matching accuracy of the joint multi-scale extraction and temporal reduction through comparisons of different temporal reduction strategies, as well as the cross-camera matching strategy based on Bag of Binary Words, and the application to several binary descriptors. We also show that XC-PR achieves similar accuracy but faster, on average, than a baseline consisting of an incremental list of spatio-temporal descriptors. Moreover, XC-PR achieves similar accuracy of a frame-based Bag of Binary Words approach adapted to our approach, while avoiding to match features that cannot be informative, e.g. for 3D reconstruction.

Contents

Published work

Journal papers

[J1] Alessio Xompero, Oswald Lanz and Andrea Cavallaro. A spatio-temporal multi-scale binary descriptor. *IEEE Transactions on Image Processing*, vol.29, pp.4362–4375, January 2020.

Conference papers

[C1] Alessio Xompero, Oswald Lanz and Andrea Cavallaro. MORB: A multi-scale binary descriptor. *IEEE Conference on Image Processing*, Athens, Greece, October 2018.

[C2] Alessio Xompero, Oswald Lanz and Andrea Cavallaro. Multi-camera Matching of Spatio-Temporal Binary Features. *International Conference on Information Fusion*, Cambridge, United Kingdom, July 2018.

[C3] Oswald Lanz, Alessio Brutti, Alessio Xompero, Xinyuan Qian, Maurizio Omologo and Andrea Cavallaro. Accurate target annotation in 3D from multimodal streams. *IEEE International Conference on Acoustic, Speech and Signal Processing*, Brighton, United Kingdom, May 2019.

[C4] Xinyuan Qian, Alessio Xompero, Alessio Brutti, Oswald Lanz, Maurizio Omologo and Andrea Cavallaro. 3D Mouth Tracking from a Compact Microphone Array Co-located with a Camera. *IEEE International Conference on Acoustic, Speech and Signal Processing*, Calgary, Canada, April 2018.

Electronic preprints are available at http://www.eecs.qmul.ac.uk/~andrea/publications.html.

Acknowledgements

First, I would like to thank my supervisor Professor Andrea Cavallaro for his continuous support and advice, his uncountable number of suggestions helped me to grow professionally and personally, and in preparation for my future. Then, I would like to thank my second supervisor, Dr. Oswald Lanz, for his valuable feedback, discussions and support during the meetings and the time spent in Italy. I would like to thank independent assessor, Dr. Miles Hansard, for his useful comments during the various stages of my Ph.D.

Most of the time in these years was often spent in CS440 (SmartCameras group) and in TeV (FBK, Italy) with many deep discussions, support, jokes, helps, and outdoor events, which made the Ph.D. life completely unique. A special thank you to Ricardo Sánchez Matilla, Riccardo Mazzon, Fabio Poiesi, Shahanawaz Amed, Massimiliano Mancini, and Lin Wang for the valuable time and help, and a big thank you to Xinyuan, Changjae, Maria, Ali, Janice, Vandana, Mohamed, Mohammad, Ashish, Yiming, Girmaw, Obaid, Tristan, Sandeep, Evangelos, Simon, Omair, Lingyuin, Raheeb, Andrea S., and Swathikiran. And a thank you to all the people that we could share part of the time in either of the two labs.

I am especially grateful to my family – Claudio, Daniela, Martina, Enrico, and relatives – that they always believed and supported me, even from far, but not too far, away. Moving abroad also means changing people you know, like friends, but some are always there, even nowadays, showing how certain friendships are solid and strong. I thank Antonio Marsico and his family, and Giuseppe Barbiera and his family; two guys that got married over these years, and honoured me as their best man during the wedding.

Living in new places also means meeting new people that play an important role outside the lab. Dancing is my personal hobby and passion, and always allowed me to meet a lot of new great people that, apart from having fun, also they supported me in the difficulties encountered during this long journey. A special thank to Sylvia, Gem, Tom, Lara, Laura, Nelson, Elijah, Ana, Fabio, Mami, Pebbles, Ovgu, Yu-Jin, Halima, Angus, Virginia, Felipe, Adrian, Josè, Chiara, Paola, Shan, Giada, Valentina. And a thank you to all the rest of the dancing people I met.

To my family, for their unconditionally support
in all my choices over all these years.

"You learn more from people
who challenge your thought process
than from those
who affirm your conclusions."
Adam Grant

List of abbreviations

ACRD	The Oxford Affine Covariance Regions Dataset
AGAST	Adaptive and Generic Accelerated Segment Test
ASV	Accumulated Stability Voting
BOLD	Binary Online Learned Descriptor
BoTW	Bag of Tracked Words
BoW	Bag of visual Words
BRIEF	Binary Robust Invariant Elementary Features
BRISK	Binary Robust Invariant Scale Key-point
BTST	Binary search Tree of Stable Tracked words
CNN	Convolutional Neural Networks
D2-Net	Joint Detector and Descriptor Network
DOAP	Descriptor Optimised for Average Precision
DSP-SIFT	Domain-Size Pooling SIFT
eSURF	Extended SURF
FAST	Features from Accelerated Segment Test
fps	Frames per second
FREAK	Fast REtinA Keypoint
GCN	Global Correspondence Network
HBST	Hamming distance embedding Binary Search Tree

LATCH	Learned Arrangements of Three patCH codes
LDB	Local Difference Binary
LF-Net	Local Feature Network
LIFT	Learned Invariant Feature Transform
LIOP	Local Intensity Order Pattern
LiST	List of Stable Tracked words
LMED	Least MEDian
mAP	mean Average Precision
MIOP	Mixed Intensity Order Pattern
MORB	Multi-scale ORB
MS	Matching Score
MST	Multi-Scale Temporal
MST-S	Multi-Scale Temporally without Stability
NN-AF	Nearest Neighbour Average F_1 score
NNDR	Nearest Neighbour Distance Ratio
OIOP	Overall Intensity Order Pattern
ORB	Oriented FAST and Rotated BRIEF
P	Precision
R	Recall
R2D2	Repeatable and Reliable Detector and Descriptor
RF-Net	Receptive Field Network
RFD	Receptive Field Descriptor

SetDesc	Set of Descriptors
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localisation and Mapping
SLS	Scale-less SIFT
SOSNet	Second Order Similarity Network
SURF	Speeded Up Robust Features
T-D	Temporally Dominant
T-DS	Temporally Dominant-Stability
TTST	Ternary search Tree of Stable Tracked words
TW	Tracked Word
VLAD	Vector of Locally Aggregated Descriptors
XC-PR	Cross-Camera Place Recognition

List of symbols

B_s	Scale adaptive margin for feature point localisation
D	Dimension of the binary descriptor
F	Number of feature points localised in an image
F_s	Number of feature points for each scale s
G	Size of the patch surrounding an interest point
$\hat{K}$	Number of frames with new localised local features
L_i	Length of a feature track i
M_i	Number of stable binary values for a spatio-temporal descriptor i (stability length)
N	Maximum number of tracked words per node

N_j	Number of tracked words stored in the j-th leaf node
R	Number of estimated residuals between binary descriptor pairs across scales
S	Number of scales of a Gaussian pyramid
$\hat{T}_k$	Number of active tracked words
W	Size of the 2D Gaussian kernel to smooth an image
d	Index of the binary test of a binary descriptor
d_j^*	Optimal partitioning index at the j-th leaf node
$g(\cdot)$	Gaussian convolutional kernel
h	Hamming distance
i	Index of a feature track (or stable tracked word)
k	Frame index
$\hat{k}$	Index of the last frame with new localised local features
k_i	Index of the frame where the feature track i terminates
n	Step to detect new features in an image sequence with MST
p_i	Normalised stability length of a tracked word i
q	Index of a query descriptor or tracked word
r	Radius of the gate for guided feature matching between two consecutive frames
t	Current frame index
t_i	Index of the frame where the feature track i is initially localised
w	Size of a block for the grid-based feature point localisation
f, g	Indexes of interest points localised in an image
s, l	Scale indexes

$\mathbf{C}_k$	Camera pose (orientation + translation) at frame k
$\mathbf{I}$	Gray-scale image
$\mathbf{I}_s$	The image $\mathbf{I}$ at scale s
$\mathbf{K}$	Camera calibration matrix
$\mathbf{R}$	Rotation matrix
$\mathbf{X}_{f,k}$	Location of the 3D point back-projected from the f-th image point at frame k
$\mathbf{b}$	Binary accumulated stability voting descriptor
$\mathbf{d}$	Descriptor representing the patch around an interest point
$\mathbf{d}_{f,s}$	Descriptor representing the patch $\mathbf{p}$ centred at the f-th feature location at scale s
$\mathbf{d}_{i,k}$	Descriptor representing the patch $\mathbf{p}$ centred at the image location of feature track i at frame k
$\mathbf{m}_i$	Vector encoding the temporal stable binary values
$\mathbf{p}$	Patch surrounding an image feature point
$\mathbf{r}_{s,l}$	Residual of two binary descriptors between any pair of scales s and l
$\mathbf{s}_{f,k}$	Depth of the f-th image point at frame k
$\mathbf{u}_d$	Pixel pair in the sampling pattern $\mathcal{S}$ for d-th comparison
$\mathbf{u}_{d,1}, \mathbf{u}_{d,2}$	Sampled pixel in the sampling pattern $\mathcal{S}$ for d-th comparison
$\mathbf{w}_i$	Temporally dominant-stability descriptor or stable tracked word
$\mathbf{w}_q$	Query stable tracked word
$\mathbf{x}$	2D image location of a local feature (interest point)
$\mathbf{x}_{f,k}$	Location of the f-th image point at frame k
$\mathbf{z}_i$	Vector encoding the temporal dominant binary values
$\mathcal{D}_i$	Set of descriptors accumulated over time and associated to feature track i

$\mathcal{D}'_i$	Set of descriptors that captures the temporal changes (instability) of the binary tests in $\mathcal{D}_i$
$\mathcal{F}$	Set of feature points localised in an image
$\mathcal{I}$	Gaussian pyramid of image $\mathbf{I}$
$\mathcal{M}$	Set of matches from one view to another view (opposite direction)
$\mathcal{N}$	Set of matches from one view to another view
$\mathcal{Q}_t$	Subset of tracked words shared along with the corresponding interest points by a camera at frame t
$\mathcal{S}$	Set of pixel pairs (sampling pattern) to compare for descriptor estimation
$\mathcal{T}_i$	Spatio-temporal feature or feature track
$\mathcal{V}$	Set of valid matches between two views
$\mathcal{W}_t$	Subset of non-active stable tracked words whose last frames are within an adaptive temporal window, at the current frame t
Λ	Maximum number of frames for view selection
Ω	Size of the window for cross-correlation based Hamming distance
α_s	Weight for each scale s when computing the cross-correlation based distance between multi-scale binary descriptors
β	Locally threshold for the binary accumulated stability voting descriptor
γ	Hamming distance threshold
$\hat{\gamma}_{i,q}$	Dynamic Hamming distance threshold between a tracked word i and a query tracked word q
δ	Threshold for Lowe's ratio test
η	Minimum number of frames before sharing tracked words across cameras
θ	Camera intrinsic parameters

λ	Scale factor to compute the scales of a Gaussian pyramid
μ	Scale offset between multi-scale descriptors of a correctly matched pair
$v_{i,k}$	Visibility of the i-th stable tracked word in frame k
χ	Ratio for minimum number of active tracked words/frame
$\pi(\cdot)$	Projective transformation of a pinhole camera model
ρ	Minimum length of a tracked word
σ	Standard deviation of the 2D Gaussian kernel to smooth an image
$\tau(\cdot)$	Triangulation function
φ	Orientation angle for a local image feature
$\psi(\cdot)$	Function that extract a patch centred at the location of a an interest point from a given image
$\oplus$	Logical XOR operator
$\wedge$	Logical AND operator
$\langle \cdot, \cdot \rangle$	Logical dot product

Chapter 1

Introduction

1.1 Motivation

In recent years, the number of devices with low cost and integrated cameras, such as micro aerial vehicles, ground-vehicles, robots, smartphones, and head-mounted displays, has increased. These devices can connect directly to each other and exchange information extracted from images and video streams acquired by the cameras to perform high-level visual tasks, such as the understanding and analysis of the environment that the devices are sensing [78] (see Figure 1.1).

Independently moving cameras can be deployed in unknown environments, where GPS measurements might be unavailable or are inaccurate, such as indoor or urban environments, or for applications that cannot be handled by a single device, such as surveillance [2], search and rescue [19, 32, 82], large 3D reconstruction via Structure from Motion or Collaborative Visual Simultaneous Localisation and Mapping (SLAM) [19, 32, 74, 82, 83, 110, 121], autonomous navigation [28, 68] and augmented reality [46]. In this context, cameras can observe the scene from different distances or viewpoints, making the matching of the content extracted from the visual stream of one camera with the content obtained by another camera (*view matching*) very challenging. This problem can limit the collaboration between the cameras and success of the task to perform. Existing vision methods extract either local or global information for each image, or temporal information from the visual stream.

Local methods describe the neighbourhood (or patch) of a localised interest point with a distinctive signature (*descriptor*), assuming that the scene contains enough texture [12, 17, 50, 55,

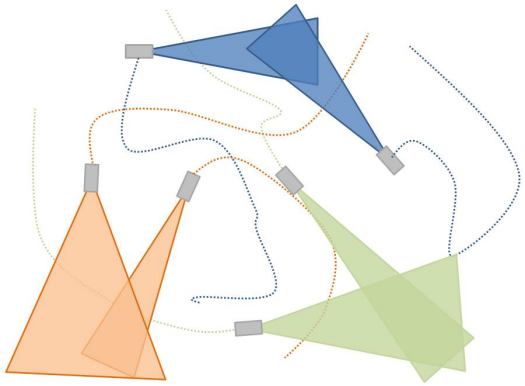


Figure 1.1: Multiple cameras freely moving in an environment (top-down view). The limited field of view and the unconstrained motions make the view matching challenging as the cameras can observe the scene from different distances and viewpoints at different instances.

77]. These *local image features* should be repeatable, distinctive, accurate, efficient, robust to image deformations, and invariant to geometric transformations [103]. *Global* methods, instead, represent the whole image with a compact representation either by directly extracting a feature vector from the image, or by aggregating the local image descriptors into a global feature vector [8, 33, 70, 90]. For example, in image retrieval [90] and *visual place recognition* [56], a query image is matched to an image in a large set using global methods that can cope with illumination, appearance, seasonal, and day-night changes [7]. This can then enable applications, such as Visual Localisation [18] or 3D reconstruction via Structure from Motion [83], that aim to localise the camera associated to the query image with respect to a previously stored map, consisting of the large set of images with corresponding reconstructed 3D scene points.

In addition to this, approaches that jointly and incrementally estimate the ego-motion (*i.e.* the camera poses over time) of a single moving device and reconstruct the surrounding 3D scene using only images as input (Visual SLAM [34, 68]) are prone to drift errors, *i.e.* the error between the estimated and the real trajectory increases over time. To self-correct inaccurate estimations, reducing temporally accumulated drifts and reconstruction inconsistencies, these approaches match the current image against the previous frames in the image sequence. As SLAM approaches are usually deployed in pre-defined paths, *e.g.* a road, camera-equipped platforms can return on the same path that was previously visited, forming a *loop*. Therefore, these approaches adopt a strategy that geometrically verifies candidate frames identified by visual place recognition to avoid false detections (*loop closure detection*). To finally correct the drift in loop, SLAM proceeds with a merging and optimisation stage.

To handle geometric variations, most of the existing loop closure detection strategies rely on local image features aggregated in compact representations for each image or in incremental structures of the sequence [33, 36, 69, 102]. While the global representation helps to determine the most similar images of a previously seen place as first step, local features are then matched between two images to geometrically validate the candidate place, *e.g.* using the epipolar constraint [38]. However, the feature similarity normally decreases with the increase of the geometric or photometric transformation, and matching ambiguities can arise when features visible in one view may be occluded in another view. This can easily occur when independently moving cameras observe the scene from different distances and viewpoints. Collaborative SLAM methods [19, 32, 82, 74, 118, 121] build on existing monocular SLAM approaches, applying the same loop closure detection strategies that, however, may be too restrictive in this context to effectively and efficiently enable the completion of a collaborative task. Therefore, an important open problem for cross-camera view matching is how to design a descriptor that is robust to severe changes in viewpoint and scale, while preserving the efficiency that is required for the real-time processing and the data to exchange due to the limited bandwidth for the communication.

Alternatively, local features can benefit from temporally accumulated information to improve their descriptiveness and robustness [99, 105]. *Local spatio-temporal features* can be extracted within a (fixed) temporal window [105], or online by tracking local image features [68, 99]. The former approach uses spatio-temporal feature detectors to localise interest points in spatial, temporal, and scale domains [48, 105] and then employs spatio-temporal descriptors to encode appearance, motion (*e.g.* optical flow), and statistics (*e.g.* image gradients) of the spatial and temporal neighbours of the interest points [105]. As the local temporal structure depends on the camera viewpoint, these features are mainly designed for in-camera tasks, such as human action recognition [35, 105], and are unsuitable for cross-camera matching with considerable geometric changes [99]. Online approaches use tracking instead to obtain the spatio-temporal features. However, in the context of multiple moving cameras, the potential of spatio-temporal features is under-investigated and the challenge is a trade-off between the descriptiveness and efficiency of the local temporal descriptor within the camera for tracking, and the repeatability and invariance to geometric differences, such as scale and viewpoint, with another view [35, 99].

In this thesis, we tackle this challenging problem of view matching between pairs of independently moving cameras. We propose generic frameworks for obtaining and matching local

features, exploiting multi-scale and temporal extraction, while preserving efficiency. We also propose to embed the spatio-temporal features in a novel approach to identify previously seen places between cameras over time (cross-camera place recognition).

1.2 Research problem

Let two cameras moving independently in an unknown environment. For each frame k, each camera acquires image $\mathbf{I}_k$. We define view matching as the problem of matching the content from the image sequence of one camera with the content from the image sequence of another camera. We do not require the image sequences of the two cameras to be synchronised, and we assume that the camera poses and calibration data are available, if needed. We also assume that the environment contains enough texture or objects such that local features can be localised in the images.

If the cameras look at the environment from different distances at time k, corners on objects lying in the scene are observed in each image with different scales, depending also on the object-to-camera distance. The problem that arises is how to localise and describe the local image neighbourhood of each corner to improve the feature matching between the two views. Our research objective is to investigate this problem from a multi-scale description approach with compact representation (*e.g.* binary) to preserve efficiency for moving cameras, as opposed to multi-scale localisation approaches and/or less efficient representations (*e.g.* histogram-based).

When the cameras are moving, the representation of each corner can be enriched with more distinctive but also redundant or unnecessary information, which is not all available in advance but is obtained frame-by-frame. Our second objective is to investigate how to encode the temporal information for compact representations associated with corners in 3D to increase the matching accuracy between cameras. In addition to this, the joint multi-scale representation and the temporal encoding will be investigated to handle viewpoint and scale differences during the motion of the cameras.

The more the cameras move around the environment, the more the number of local (temporal) features to match between the cameras, hence increasing the amount of time required to match across cameras. Our third research objective is to investigate a strategy to organise the temporal features so that searching and matching can be performed in a more efficient way, as well as to select when and what to share between the cameras, enabling a continuous and online approach

to retrieve similar views across the cameras.

1.3 Contributions

Given two cameras that independently move in non-planar scenes, acquire image sequences and directly interact to each other, our aim is to design compact local features that can cope with challenges, such as scale and viewpoint differences, while improving the accuracy to match the features and identify previously seen places. We propose, in this thesis, methods under two cases: uncalibrated and unsychronised cameras, and calibrated cameras to leverage the 3D information for validation of the features. We evaluate the accuracy of feature matching between image pairs or across cameras that acquire short image sequences (*e.g.* 100 frames), whereas we assess the accuracy of recognising previously seen places across cameras that acquire longer sequences (*e.g.* $>$300 frames).

The main contributions of the thesis are the following:

1. An image-based multi-scale binary descriptor to cope with large scale variations between images acquired at different distances. The descriptor describes an image patch at different scales of an image pyramid using an oriented sampling pattern of intensity comparisons in a pre-defined set of pixel pairs. A scale-aware Hamming distance computed between descriptors of each view allows the identification of the best matches at descriptor scales that may differ from the scale where the local feature was initially localised. This improves the matching accuracy compared to existing binary features that describe local image features only at the detection scale. [C1]

2. A local spatio-temporal binary feature to cope with viewpoint changes between short visual streams acquired by uncalibrated and unsynchronised moving cameras. The feature accumulates temporal information by tracking a local image feature within each view and extracting a sequence of binary vectors that encode the intensity comparison of pixel pairs for each image patch. The sequence of vectors is then reduced to a compact fixed-length representation by selecting the temporally dominant binary values. This representation is also complemented by a second vector that identifies intensity comparisons that are temporally unstable, and acts as a selector for the first vector to ignore the corresponding binary values when matching the features between views. A selective weighted Hamming distance computed between feature pairs allows the estimation of the best matches increasing

the number of correct matches. [C2]

3. A multi-scale temporal feature that captures appearance variations of a 3D scene point as observed by a calibrated moving camera to cope with severe scale and viewpoint changes in non-planar scenes. This feature builds on the multi-scale binary descriptor and the local spatio-temporal feature of previous contributions. To obtain feature point trajectories, localised interest points are tracked with a pyramidal local search, to increase the lifespan, and then validated through 3D reconstruction. The localisation is augmented with a suppression strategy that increases the scale-invariance and leads to a more desirable feature distribution. The appearance variations are thus encoded with binary vectors extracted at multiple scales of an image pyramid, and then temporally reduced into a multi-scale descriptor that identifies the most frequent and stable binary values over time for each scale. To find correspondences between views, a scale-aware weighted Hamming distance is computed between descriptors of each view. To validate spatio-temporal features across views, we contribute an evaluation method that extends that of local image features and uses multi-view geometry. The multi-scale temporal feature is generic for several image-based binary descriptors and improves the matching performance compared to alternative temporal reductions. [J1]

4. A novel cross-camera place recognition approach that selects distinctive descriptors from binary features observed in multiple frames and effectively identifies informative features to share across cameras. This approach forms, for each camera independently, stable tracked words that are obtained by associating binary features and temporally compressing their accumulated descriptors to a fixed-length representation. As the previous contribution, this representation preserves the most persistent values, which are more robust to temporal changes occurring while a local feature is tracked. However, features are extracted and reduced at a single scale, and the cameras are uncalibrated. Therefore, this approach does not validate features through 3D reconstruction. As the number of tracked words grows over time, matching them across cameras with a linear search may become computationally intractable. To enable efficient searching and matching, we insert tracked words from automatically selected frames into a hierarchical structure, an adaptive tree of stable tracked words. We formulate the structure as a search tree that adapts over time through the insertion of new tracked words and the removal of short tracked words. When the number of

tracked binary features is reduced due to the view change caused by the camera motion, a camera localises new binary features and updates the hierarchical structure. The camera then shares a subset of tracked words along with the image coordinates selected from the frame with the largest number of corresponding binary features, within an adaptive temporal window. The approach finally recognises a place within the camera that receives the query tracked words by identifying and geometrically validating a previous frame with the largest number of matched tracked words.

1.4 Organisation of the thesis

This thesis is organised as follows.

Chapter 1: we introduce and formulate the view matching problem between pairs of cameras that independently move in an unknown environment, and we list the contributions.

Chapter 2: we review the literature of local image features, such as histogram-based, binary and deep learning based methods, and we discuss methods that handle scale variations as well as spatio-temporal features for viewpoint differences. We then review previous works on visual place recognition and loop closure detection, and conclude the chapter by discussing datasets and performance measures used for local image features and loop closure detection.

Chapter 3: we present three features: a novel multi-scale binary descriptor for handling scale differences between images, a spatio-temporal feature for improving matching accuracy under viewpoint differences, and a novel multi-scale temporal feature with 3D reconstruction to cope with both scale and viewpoint differences in non-planar scenes. For each feature, we introduce a corresponding distance for the matching strategy and we conclude by discussing the methods compared to the literature.

Chapter 4: we present the novel cross-camera place recognition framework, illustrating the online formation of the binary tracked words and their organisation in an incremental and adaptive tree. We then introduce the selection and sharing of tracked words between views to identify a previously seen place via tracked words search and matching within the tree, followed by a geometric verification.

Chapter 5: we evaluate the three proposed compact features and the cross-camera place recognition framework. We evaluate the proposed multi-scale binary descriptor on standard image matching datasets. We also evaluate spatio-temporal and multi-scale temporal features

on collected and annotated scenarios using short image sequences. We introduce a performance evaluation for the spatio-temporal features that extends the evaluation used for local image features. We also show that the proposed features are generic for different local image binary descriptors. We then evaluate the accuracy and speed of the proposed cross-camera place recognition framework by comparing different strategies to organise features within each camera.

Chapter 6: we summarise the methods and the achievements presented in this thesis, and we discuss the future work.

Appendix A: we present our dataset that consists of four scenarios with natural and man-made environments where multiple hand-held cameras are deployed, while independently moving around, for cross-camera place recognition. We also present the annotation procedure of the dataset and the performance measures to evaluate the cross-camera place recognition.

Appendix B: we present alternative investigated strategies and their validation for matching multi-scale binary descriptors across scales.

Appendix C: we present an approach to reduce the multi-scale temporal binary descriptor across scales based on the principle of the accumulated stability voting, and we evaluate and discuss five variants in comparison with the original descriptor without reduction.

Chapter 2

Literature review

2.1 Introduction

In this chapter, we review the literature on local image features, spatio-temporal features and visual place recognition for addressing the problem of view matching. In Section 2.2, local image features are categorised into histogram-based, binary, and deep learning based approaches. While most of the features are designed to be invariant to several geometric transformations between images, we discuss, in detail, how scale variations have recently been handled. In Section 2.3, we introduce and categorise spatio-temporal features according to their extraction approach, such as those based on local volume and those obtained by tracking local image features online. Visual place recognition is discussed in Section 2.4, focusing on different ways to represent a place and on different loop closure detection strategies. We then review the datasets (Section 2.5) and performance measures (Section 2.6) commonly used in the literature for image matching.

2.2 Local image features

Descriptive local image features are fundamental for a number of applications, including Structure from Motion [83, 91], Visual SLAM [46, 68], Object Retrieval [55], Image Stitching [16] and Image Retrieval. A local feature describes the neighbour of a interest point (*e.g.* a corner) that is localised by a detector, such as Harris' [37] or Difference of Gaussians [55]. The interest point can be localised at the resolution of the original image scale or at a coarser scale using an image pyramid [103]. To localise interest points in an image, response functions that determine,

for example, the cornerness [37, 55, 59, 76, 77] or spatial constraints can be applied to retain only valid candidates (non-maxima suppression strategies). A comprehensive review of local image features can be found in Csurka *et al.* [22], while Tuytelaars and Mikolajczyk [103] provide a detailed review of local image detectors. Local image features may be described using histogram representations of gradients or intensities of local patches (histogram-based features); binary descriptors; or Convolutional Neural Networks (CNN) operating on patches or on the whole frame (*i.e.* deep learning based features). Table 2.1 summarises the approaches discussed in this section and Section 2.3.

2.2.1 Histogram-based features

Histogram-based features include Scale Invariant Feature Transform (SIFT) [55] and its variants [8, 13, 14, 44, 61], Speeded Up Robust Features (SURF) [12], Daisy [98], Local Intensity Order Pattern (LIOP) [107], Overall Intensity Order Pattern (OIOP) [106] and Mixed Intensity Order Pattern (MIOP) [106]. SIFT [55] and its variants [8, 13, 26, 113] describe statistics of the patch and accumulate gradient orientation information. To account for in-plane rotations, SIFT computes the dominant orientation of the patch using the gradient orientation information, and extracts the descriptor after applying the transformation. After observing that the Hellinger distance is less sensitive to large bin values than the Euclidean distance (or L2-norm) when comparing histograms, RootSIFT [8] transforms the SIFT descriptor in such a way that computing the Euclidean distance between two RootSIFT descriptors is equivalent to computing the Hellinger distance between two SIFT descriptors. This transformation is valid and generic for all histogram-based descriptors. SURF [12] approximates the gradient with responses of Haar wavelets to increase the computational efficiency during the feature extraction. Daisy [98] estimates convolutional oriented maps for each pixel with Gaussian filters, and has a similar invariance to SIFT but a better efficiency for dense matching. LIOP [107], OIOP [106] and MIOP [106] rank pixels in a patch according to their intensity value, which is assigned to an intensity bin (ordinal cluster). This design makes the three features rotationally invariant, that means there is no need to compute the dominant orientation as extra step before describing the patch. LIOP [107] encodes the local ordinal information of each pixel by mapping the quantised intensities of corresponding neighbouring sampling points to a decimal code via a look-up table. OIOP [106], instead, encodes the overall ordinal information by linearly combining the quantised values. The normalised histogram of the LIOP and OIOP codes are then computed for each ordinal cluster

Table 2.1: Local image and spatio-temporal features. Gray cells denote properties not handled by the method. Methods proposed in this thesis and corresponding properties are also shown at the end of the table. KEY – Ref: reference; Rot: rotation; Dist: distance used for matching descriptors; Uns: unsupervised; MS: multi-scale; SI: scale invariant; V: space-time volume; T: tracking; CNN: convolutional neural network; concat.: concatenation; L: learnt; E: Euclidean; H: Hamming; W: weighted Hamming; DOG: difference of Gaussians; GP: Gaussian pyramid; FED: fast explicti diffusion; RI: rotation invariant; IC: intensity centroid; LG: local gradient; DD: data dependent; F: floating point; B: binary; $\times$: descriptor dimension resulting from a concatenation operation or a set representation; STIP: spatio-temporal interest point [48]; Ran.: random. NN: nearest neighbour; PCA: principal component analysis.

	Detection			**Description**							
Ref Method	**App.**	**Scale** MS	SI	**Approach**	**Rot**	**Scale** MS	**Time** V	T	**Dimension**	**Stor**	**Dist**	**Uns**
[117] DeepCompare				CNN with pairs of labelled patches	DD				256	F	L	
[88] DeepDesc				CNN with pairs of labelled patches	DD				128	F	E	
[10] TFeat				CNN with triplets of labelled patches	DD				128	F	E	
[64] MR CNN				CNN with scaled patches (3 layers)	DD	✓			128	F	E	
[96] L2-Net				CNN with NN in a batch of patches	DD				128	F	E	
[62] HardNet				maxim. of nearest pos. and neg. dist. in L2-Net	DD				128	F	E	
[41] DOAP				listwise ranking-based optimis. in L2-Net	DD				128	F	E	
[57] GeoDesc				perspective geometric constraints in L2-Net	DD				128	F	E	
[97] SOSNet				second order similarity constraint in L2-Net	DD				128	F	E	
[6] FCRN-PDN	L		✓	Scale branches det. + labelled patch triplets	DD				128/256	F	E	
[116] LIFT	L		✓	CNN with quadruplets of patches	L				128	F	E	
[72] LF-Net	L		✓	CNN with pair of images + depth	L				256	F	E	
[86] RF-Net	L		✓	Receptive feature maps for LF-Net	L				128	F	E	
[24] SuperPoint	L	✓		CNN with homographic adaptation	DD				256	F	E	✓
[27] D2-Net	L	✓		joint detector/descriptor with L2-Net	DD				512	F	E	
[73] R2D2	L	✓		Repeatable and reliable feature maps for joint det./desc. with modified L2-Net	DD				128	F	E	✓
[55] SIFT	DoG		✓	gradient orientations in a regular grid	LG				128	F	E	✓
[98] Daisy	Dense			convolved orientation maps					200	F	E	✓
[107] LIOP				local ordinal intensities	RI				144	F	E	✓
[106] OIOP				overall ordinal intensities	RI				256	F	E	✓
[106] MIOP				concat. of LIOP with OIOP + PCA	RI				128	F	E	✓
[39] SLS	DoG		✓	linear subspace of SIFTs	LG	✓			8256	F	E	✓
[26] DSP-SIFT	DoG		✓	pooling of SIFTs across scales	LG	✓			128	F	E	✓
[113] ASV	DoG		✓	SIFTs/LIOPs stability across scales	LG	✓			128/144	F	E	✓
[85] 3D-SIFT	Ran.			3D gradient orientations	LG		✓		256/2048	F	E	✓
[105] HOG3D	STIP	✓		3D-SIFT with polyhedrons	LG		✓		960	F	E	✓
[99] Daisy-3D	Dense	✓		concat. of Daisys with optical flow				✓	7×136	F	E	✓
[17] BRIEF				random set of pixel pairs					128/256/512	B	H	✓
[77] ORB	GP	✓		learnt set of pixel pairs	IC				256	B	H	✓
[50] BRISK	GP		✓	deterministic set of pixel pairs	LG				512	B	H	✓
[114] LDB				learnt set of sub-patch pairs	IC				256	B	H	
[5] A-KAZE	FED		✓	Scale-based sub-sampling of the LDB grids	RI				265	B	H	
[51] LATCH				learnt set of sub-patch triplets	IC				128/256/512	B	H	
[101] D-BRIEF				linear comb. of box/Gaussian filters	DD				32	B	H	
[100] BinBoost				learnt set of hash functions (boosting)	DD				64	B	H	
[29] RFD				selected receptive fields + learnt thresholds	DD				293/598	B	H	
[11] BOLD				online selection of stability bits					512	B	W	✓
[53] DeepBit				CNN with min quantis. + max entropy loss	RI				256	B	H	✓
[115] CDBin				lightweight CNN with triplet loss	RI				256	B	H	
[95] GCNv2	L	✓		Lightweight CNN for motion estimation					256	B	H	
[68] LMED				ORB selection over time	IC			✓	256	B	H	✓
[52] STB				optical flow and temporal gradients encoding	IC		✓	✓	188	B	H	✓
[C1] MORB	GP	✓		set of ORBs across scales	IC	✓			8×256	B	H	✓
[C2] T-DS				temporally reduced ORBs (centroid + stability)	IC			✓	512	B	W	✓
[J1] **MST**	GP		✓	set of temporally reduced ORBs across scales	IC	✓		✓	5×512	B	W	✓

and concatenated to form the descriptor. MIOP [106] exploits the complementary information between the two descriptors at a reduced dimensionality (128 vs. 144/256 bytes) by applying principal component analysis to the concatenation of LIOP and OIOP. LIOP, OIOP and MIOP outperform SIFT and Daisy [106], with MIOP being the best performing [106]. Histogram-based features are vectors whose dimensionality usually varies between 128 and 256 but can be even larger (*e.g.* 8256 in case of Scale-Less SIFT (SLS) [39], see Table 2.1). Each element is stored as a floating value (bytes) to cover the high possible range of values or the real values when normalised. Histogram-based features are thus matched using the L2-norm as distance metric; however, for applications with time constrained and low storage requirements, these features may not be sufficiently compact and computing the L2-norm may not be so efficient, especially when the number of features to match increase.

2.2.2 Binary features

Binary features aim to describe the local image area of a detected interest point with a compact vector of binary values for low-storage and high efficiency requirements, maintaining good performance in terms of robustness to geometric and photometric transformations, as opposed to other descriptors, such as SIFT [55] or SURF [12]. Most of the existing binary features use corner detectors, such as Features from Accelerated Segment Test (FAST) [76] or Adaptive and Generic Accelerated Segment Test (AGAST) [59], specifically designed to be both repeatable and efficient for real-time applications, and propose descriptors that result from hash or projection functions followed by thresholding [29, 92, 100, 101], or from tests on pre-defined sampling patterns that are defined deterministically [50] or probabilistically [17], or learnt [3, 51, 77, 114]. Moreover, feature matching becomes more efficient with binary features, as the binary vectors can be compared using the low-level XOR bitwise operator and the *L2*-norm can be replaced by the Hamming distance.

Examples of binary features generated from comparisons of intensity values of pre-defined pixels pairs in a sampling pattern are Binary Robust Invariant Elementary Features (BRIEF) [17], Oriented FAST and Rotated BRIEF (ORB) [77], Fast REtinA Keypoint (FREAK) [3] or Binary Robust Invariant Scale Key-point (BRISK) [50]. Distinctiveness can be increased by extending comparisons to statistics of small window pairs [114] or triplets (with one small window acting as anchor) [51]. BRIEF [17] randomly samples the tests from a Gaussian distribution. BRISK [50] uses a deterministic sampling pattern, where the points lie on appropriately scaled concentric cir-

cles. FREAK [3] is inspired by the human retina, and uses a circular pattern with higher density near the centre, *i.e.* the location of the corner point in the image. ORB and FREAK learn the sampling pattern in an unsupervised way, with a variance-correlation bit selection strategy. To perform a faster matching, FREAK also exploits the coarse-to-fine structure of the descriptor by simulating the saccadic search of the humay eye movements and using a cascade of comparisons. Unlike previous methods, Local Difference Binary (LDB) [114] and Learned Arrangements of Three patCH codes (LATCH) [51] minimise the distance between pre-annotated matching interest points. A-KAZE [5] modifies the LDB descriptor by sub-sampling its grids as a function of the scale where each feature point is localised, making the descriptor robust to changes in scales. Unlike LDB, A-KAZE provides rotation-invariance and exploits the novel non-linear scale-space based on the fast explicit diffusion (FED) scheme to speed up the feature localisation. However, the efficiency gained with these binary comparisons comes at the cost of a reduced matching accuracy and robustness to geometric transformations and photometric variations. For example, Binary Online Learned Descriptor (BOLD) [11] shows that binary values can change (instability) when extracting the descriptor under small geometric variations (*e.g.* scale or affine), and addresses this problem by selecting the most discriminative tests, after quantifying their stability. The stability flag for each binary test is encoded as an additional binary vector.

Examples of binary descriptors based on hash or projection functions are LDA-Hash [92], D-BRIEF [101], Binboost [100], and Receptive Field Descriptor (RFD) [29]. To obtain the binary representation, LDA-Hash [92] applies discriminative hash functions to SIFT descriptors followed by a threshold. D-BRIEF [101] projects the patch intensities to a compact binary representation using a linear combination of box or Gaussian filters. Binboost [100] learns a set of hash functions that are the binary response of a boosting strong classifier built as a linear combination of weak classifiers. RFD [29] learns a binary descriptor by first selecting the set of most discriminative receptive fields, defined as the aggregation of low-level filter responses within a patch, and then binarising the responses with learnt thresholds for each receptive field.

These representations can outperform even histogram-based features (*e.g.* SIFT) in image or patch matching problems, but are less efficient than early binary features, and unsuitable for matching in time-constrained applications.

Finally, DeepBit [53] is a CNN-based approach that learns a binary descriptor in an unsupervised manner: an image patch and its geometrically transformed version are given as input

to a Siamese network to learn a set of projection functions to provide invariance to the transformations; enforce minimal quantisation error between the real-valued deep feature and the binary code to increase the descriptiveness (quantisation loss); and evenly distribute the binary code to maximise the information capacity (entropy) for each bin (even-distribution loss). CDbin [115], instead, uses a lightweight CNN to reduce the number of parameters and increase the efficiency of training and testing, outperforming other state-of-the-art binary descriptors. In addition to quantisation and even-distribution loss, CDbin uses a supervised triplet loss to increase the discriminative power and a correlation loss to reduce the correlation among different bits.

2.2.3 Deep learning based features

After the success of deep learning in tasks such as image classification [47], recent methods adopt CNNs for learning discriminative local features, and aim to achieve the same performance, or outperform, histogram-based and binary features.

Patch-based CNN features learn to discriminate correct and incorrect matches with supervised training. Examples include DeepDesc [88], DeepCompare [117], TFeat [10], Multi-resolution CNN (MR-CNN) [64], L2-Net [96], HardNet [62], GeoDesc [57], Descriptor Optimised for Average Precision (DOAP) [41], and Second Order Similarity Network (SOSNet) [97].

DeepDesc [88] and DeepCompare [117] train a Siamese network with pairs of annotated patches to push away incorrect patches and to move corresponding patches closer on a distance metric, such as Euclidean, Hamming, or learnt. To reduce overfitting, TFeat [10] extends this network to triplets (anchor, positive sample, negative sample) and uses hard negative mining with anchor and positive samples swap to increase the distinctiveness of the resulting descriptor. To improve scale invariance, MR-CNN [64] learns a descriptor using image patches scaled at three resolutions as input to a three-stream Siamese network. However, TFeat outperforms MR-CNN in patch and image matching, as well as in efficiency.

L2-Net [96] goes beyond the pairwise or triplet samples for training and optimises the relative Euclidean distance among many descriptors in a batch of patches to better resemble the nearest neighbour search without caring about the magnitude of the distance. A progressive sampling strategy allows L2-Net to efficiently access a large number of patch pairs and thus to learn a descriptor that outperforms previous approaches on patch and image matching tasks. HardNet [62] learns a more discriminative descriptor by simplifying the optimisation function of L2-Net and maximising the distance between the closest positive and the closest negative sample pairs in the

batch (inspired by the Lowe's ratio test for SIFT [55]). GeoDesc [57] enforces geometric constraints, such as patch similarity and image similarity, that measure the difficulty with respect to perspective changes of data obtained with multi-view reconstructions. These constraints allows GeoDesc [57] to construct batches of patches with in-batch hard samples during training, with the advantage of avoiding overfitting and sampling training data that is more consistent with realistic and complex testing scenarios. SOSNet [97], instead, extends the optimisation function with a regularisation term that uses second order similarity between pairs within a batch of patches to capture more structural information, while being robust to deformations and distortions. Differently, DOAP [41] optimises the nearest neighbour matching stage with a performance measure, *i.e.* the Average Precision, commonly used for ranking-based retrieval problems, and thus moving from a pairwise to a listwise ranking formulation.

To improve the matching performance of all these approaches, data augmentation can be used to increase the volume and include more changes between viewpoints that may not be captured in the available training data, also helping to reduce overfitting. Data augmentation consists of transformations, such as flip, rotation, crop, translation, scale, or additive Gaussian noise, applied to the training data, offline or on-the-fly in a mini batch.

Image-based CNN features learn to localise and describe interest points on the whole image. Examples include Fully Convolutional Recursive Network - Patch Descriptor Network (FCRN-PDN) [6], Learned Invariant Feature Transform (LIFT) [116], Local Feature Network (LF-Net) [72], Receptive Field Network (RF-Net) [86] Superpoint [24], Joint Detector and Descriptor Network (D2-Net) [27], Repeatable and Reliable Detector and Descriptor (R2D2) [73], and Global Correspondence Network (GCN) [95].

FCRN-PDN [6] learns to detect scale-invariant keypoints using a multi-scale branching mechanism within a fully convolutional recursive network. To assign a descriptor to the extracted patches, a second CNN is used that, similarly to TFeat, is trained with a triplet loss. Each network is trained independently in a self-supervised manner with data collected through Structure from Motion with aerial images at different scales. LIFT [116] uses an end-to-end network to replicate the SIFT pipeline by learning detector, orientation estimator and descriptor in cascade, starting from the descriptor stage. For learning the descriptor, LIFT extends TFeat to quadruplets, including an image patch with non distinctive information. The training is based on corresponding patches extracted using SIFT features within a Structure-from-Motion framework, and,

therefore, LIFT cannot learn where SIFT fails [72]. Instead of relying on supervised labelled data, LF-Net [72] uses ground-truth camera poses and depth images to improve the learning of the end-to-end feature extraction pipeline. RF-Net [86] improves LF-Net using receptive feature maps for estimating scales-space, orientation, and score maps, and proposing a modified loss function. RF-Net localises a pre-defined number of interest points by retaining those with the highest score, resulting from the three maps. During training, the modified loss function accounts for the error between the score map of an input image and a warped score map of a second image with geometric or photometric variations with respect to the input image. Moreover, both orientation and scale maps are considered in the loss function to select patches that allows to minimise the distance from the estimated descriptors. Then, RF-Net uses L2-Net for learning the descriptor for each extracted patch, and modified the hard loss of HardNet [62] by masking those in-batch matching pairs that lead to ambiguities.

Unlike previous approaches that separate the learning of detector and descriptors under the *detect-then-describe* paradigm [27], Superpoint, D2-Net and R2D2 jointly learn detector and descriptor, showing the advantage of localising interest points that are more suitable for matching in addition to be repeatable. Superpoint [24] is a self-supervised approach that estimates interest point locations and associated descriptors directly on raw input images, assuming that the model is a homography. However, training on synthetic images or real images with affine transformations does not guarantee the applicability of SuperPoint to image pairs with wide-baseline. D2-Net [27] shows how the feature maps extracted by a CNN can be simultaneously interpreted as local descriptors and detection maps, to postpone the decision of the locations of the interest points. To jointly optimise both description and detection, D2-Net thus modifies the triplet margin rank loss with a weighted average that accounts for the detection scores. R2D2 [73] uses a L2-Net network with one output containing the dense descriptor for each pixel of an image, and two outputs for obtaining a confidence map on the repeatability and a confidence map on the reliability (*i.e.* the distinctiveness) for each pixel of an image. While the repeatability map is learnt as a self-supervised task, where positions of the local maxima are enforced to be covariant to viewpoint and illumination changes between two images, the reliability map exploits the advantage of training the network by optimising a global measure, *e.g.* a differentiable approximation of the Average Precision, with a ranking list within a group of pairs of image patches [41]. Learning both detector and descriptor shows to perform comparably well, or even better, than

disjoint methods, where either the detector or descriptors are learnt, or both, under challenging conditions, such as geometric and photometric changes. Moreover, the extraction time is reduced compared to only learned descriptors based methods, which require the processing of each patch independently and sequentially. However, the efficiency is still limited compared to, for example, binary descriptors, and the localisation of the interest points is less accurate than SIFT-like methods [27]. To achieve efficient feature extraction, GCNv2 [95]), a lightweight version of Global Correspondence Network (GCN) that, similarly to Superpoint, detects and describes interests points for camera motion estimation, provides an efficient approach to extract binary features, aiming to replace the ORB features in ORB-SLAM [68], while still running in real-time on an embedded device (*e.g.* 20 Hz on Jetson TX2). Even though this approach achieves higher accuracy than classical methods, the capability to handle severe geometric and photometric differences has not yet been validated.

Despite the fact that CNN-based features were initially found to improve performance with respect to SIFT or other hand-crafted features [10, 89, 116, 117], recent evaluations and benchmarks show that there is still no clear evidence indicating that learnt features outperform hand-crafted features, for example in 3D reconstruction [9, 11, 84]. Moreover, rankings of the methods are inconsistent across benchmarks, as evaluations highly depend upon the applications and the performance measures used [9]. For example, CNN-based descriptors outperform both histogram-based and binary descriptors on standard patch verification, image matching, or patch retrieval datasets [9, 61, 111]. However, CNN-based descriptors require large training data that can limit the generalisation across datasets and applications, whereas advanced histogram-based methods, such as RootSIFT [8], are still preferable for their robustness to geometric challenges and can be applied without requiring any training or re-training [83, 84, 22]. The computation time of extracting and matching both CNN-based and histogram-based descriptors, however, makes them less suitable for time-constrained applications, unless GPU accelerations are used (*e.g.* GPU-SIFT [112] or TFeat [10]). CNN-based binary features, such as the Binary L2-Net, DeepBit, or GCNv2, can also be extracted to obtain the same matching efficiency as standard binary features, however an efficient extraction still depends on the number of parameters of the network and the availability of GPU acceleration.

2.2.4 On handling scale variations

To address scale differences between images, histogram-based and binary features usually localise interest points or blobs at multiple scales of an image pyramid (*e.g.* Gaussian pyramid, Difference of Gaussians, Non-linear diffusion) [5, 50, 55, 60, 77, 103].

Descriptors of multi-scale approaches, extracted *at the scale* where the interest point is localised, can be inaccurate when matching across images with severe scale variations [50, 55, 77, 113]. Moreover, redundancies and ambiguities may arise if interest points are localised independently for each scale (*e.g.* ORB [77]), and can be avoided by suppressing non-maxima across scales [103] (*e.g.* SIFT [54] or BRISK [50]).

Descriptors can also be extracted at *multiple scales* of a Gaussian pyramid to capture multi-scale information of an interest point [26, 39, 113]. Coarser levels allow one to distinguish locally repeated patterns, whereas finer levels capture subtle changes, helping to discriminate nearby points [64]. The Scale-less SIFT (SLS) descriptor [39] approximates SIFT descriptors sampled at multiple scales with a linear subspace. Domain-Size Pooling SIFT (DSP-SIFT) [26] aggregates SIFT descriptors by pooling the values of each bin across scales. Accumulated Stability Voting (ASV) [113] thresholds the absolute difference between SIFT/LIOP descriptors of any pair of scales and accumulates the relative stability values into a compact representation. ASV selects one or multiple thresholds based on the principle of maximum entropy. In a second stage, ASV uses a further threshold to obtain a binary representation. However, SLS, DSP-SIFT, and ASV inherit the same limitations of the histogram-based descriptors, and are inadequate under severe viewpoint changes.

As the patches of localised interest points are provided in advance, most of patch-based CNN features can use two-stream architectures, where the input of the second stream is obtained by cropping and resizing the central part of original patches, to handle scale variations [10, 96, 117]. Multi-resolution CNN [64] down-samples and resizes the patch with bilinear interpolation twice to extract and then concatenate learned feature vectors at three resolutions. Image-based CNN features, instead, mostly handle the scale differences as traditional approaches [55, 60, 77] and localise interest points in scale-space adopting an image pyramid directly in testing phase, *i.e.* single feed-forward of the network when processing a single image, or embedding the scale-space detection in the network during training. LIFT [116] applies the detector independently to each scale of an image pyramid to obtain score maps at multiple resolutions and then a non-maxima

suppression strategy across scales [55] retains a set of feature locations. After re-sampling the image at half and double resolutions, D2-Net [27], instead, localises and describes features in a coarse-to-fine way across the scales of the image pyramid and propagates features between scales to avoid duplicates by masking. Unlike LIFT and D2-Net, LF-Net [72], embeds the scale-invariant detection of the interest points directly in the network and during the training phase. Rather than obtaining score-maps from the original image, LF-Net applies independent filters to multiple resized versions of the feature map generated by the first convolutional network block. Non-maxima suppression is thus performed independently for each score map and then to the aggregation of the resulting maps after resizing, so that a final scale-space score map is obtained and a set of interest points is retained. SuperPoint [24] does not use a scale-space formulation but, during training, applies the detector network to multiple instances of the original image after different homographic transformations that can also include the scaling operation. Score maps are then un-warped and aggregated to retain the location of the interest points with highest scores.

2.3 Spatio-temporal features

To improve the descriptiveness and robustness, local features can benefit from temporally accumulated information [99, 105]. In this section, we briefly overview spatio-temporal features that can be extracted within a (fixed) temporal window [105] or by tracking local image features [68, 99].

2.3.1 Local volume based features

Local spatio-temporal features are used for object and scene recognition, human action recognition [35, 105], video matching and retrieval [4], and wide baseline reconstruction [99].

Spatio-temporal feature detectors localise interest points in spatial, temporal, and scale domains [48, 105]. Examples include Harris3D [48], Cuboid [25], Hessian [109], and dense sampling [105]. These detectors find space-time interest points given by local maxima of a response function, such as the Harris response [37] for Harris3D, the Gabor filters-based response for Cuboid, and the Hessian saliency measures for Hessian. Harris3D and Hessian are an extension of the space-time domain of the Harris [37] and SURF [12] detectors. Dense sampling does not search for local maxima of a response function and, instead, defines the location of the interest points in a regular 5-dimensional grid, which accounts for space, time, spatial scale and temporal

scale, with a 50% overlap between sampled 3D patches [105].

Spatio-temporal descriptors encode appearance, motion (*e.g.* optical flow), and statistics (*e.g.* image gradients) of the 3D patches surrounding an interest point [105]. Examples include Cuboid [25], HOG/HOF [49], HOG3D [45], Extended SURF (eSURF) [109], and 3D-SIFT [85]. Cuboid computes the gradient for each pixel followed by principal component analysis to reduce the dimension of the feature vector. After dividing the 3D patch into smaller volumetric cells, HOG/HOF computes, for each cell, normalised histograms of spatial gradient (HOG) and normalised histograms of optical flow (HOF), each with a fixed number of bins, and concatenates them to form a single feature vector. 3D-SIFT and HOG3D extend to the spatio-temporal domain the quantisation of the histogram of gradients used in SIFT. 3D-SIFT represents the gradients in polar coordinates and quantises them in histograms by meridians and parallels. This solution leads to singularity problems near the poles [45]. HOG3D overcomes this issue by using polyhedrons and projections of the gradient vectors onto the axes that connect the centre of the polyhedron to the centre of each face of the polyhedron. eSURF extends the SURF descriptor by representing each cell of the 3D patch with a weighted sum of uniformly sampled responses of Haar wavelets.

As the local temporal structure depends on the camera view, these volume-based features are mainly designed for in-camera tasks and are unsuitable for matching across cameras with considerable viewpoint changes [99].

2.3.2 Online tracking based features

In applications such as Visual SLAM [68], Collaborative SLAM [32, 74, 81], Structure from Motion [83] or stereo reconstruction [99], local features are extracted independently for each image and matched/tracked in multiple views. Binary features, such as ORB [77], are preferred for real-time applications because these features are more compact, faster to extract and match, and can achieve a good accuracy in image feature matching benchmarks compared to the more complex SIFT features [42, 68].

Daisy-3D [99] is a spatio-temporal description for dense 3D reconstruction with a wide baseline stereo camera in the presence of non-rigid objects and occlusions. Daisy-3D captures the temporal evolution of the spatial structure of an interest point by tracking dense 2D Daisy features [98] with optical flow priors, and concatenates the temporal descriptors. Spatio-temporal features are then matched between cameras by computing an average distance of sub-descriptors

within a small window, followed by a global optimisation to enforce spatio-temporal consistency for depth estimation. The dimension of the temporal descriptors is large and therefore the matching of Daisy-3D features is computationally expensive. Moreover, to deal with dynamic objects in the scene, Daisy-3D assumes that the cameras are synchronised.

Online approaches such as ORB-SLAM [68] and STB [52], instead, obtain spatio-temporal features by tracking local binary features (*e.g.* ORB [77]). ORB-SLAM reduces the spatio-temporal feature to a compact representation by selecting the ORB descriptor from the sequence of descriptors with the least median Hamming distance from all the other ORB descriptors. Moreover, ORB-SLAM uses Bag of (Binary) Words [33] to match instances of ORB descriptors without exploiting information from the spatio-temporal feature. STB [52] encodes as binary representation the trajectory information as well as the horizontal and vertical components of the temporal gradient of a local spatio-temporal volume. Dense viewpoint- and illumination-invariant descriptors from models obtained with dense SLAM systems can be learned from RGB-D data [80] for indoor or well-structured scenes. However, the underlying SLAM system may fail outdoors due to inaccurate or incomplete depth information. Nevertheless, their performance decreases under severe geometric changes, such as scale and viewpoint, which typically occur when multiple cameras move freely.

2.4 Visual place recognition

Visual place recognition approaches can be adopted for loop closure detection in single moving cameras, for multi-session mapping and visual localisation when an existing map of reconstructed 3D points and corresponding views are already available, and for cross-camera localisation in collaborative system with multiple moving cameras. While the main challenges for loop closure detection and collaborative systems are viewpoint and scale differences, illumination changes and dynamic environments (*i.e.* presence of moving objects), for multi-session mapping and visual localisation, seasonal changes and day/night differences also become important. Moreover, collaborative systems are highly affected by severe geometric differences due to the unconstrained and simultaneous motions of the cameras, while loop closure detection approaches imposes similar viewpoints, assuming that the camera returns on a previous portion of the trajectory, *e.g.* on a road.

We briefly review and compare existing visual place recognition approaches that can be used

Table 2.2: Comparison of visual place recognition approaches and their characteristics as applied in loop closure detection for a single moving camera and in collaborative system with multiple moving cameras. KEY– Ref.: reference; Desc.: descriptor; Bin.: binary; Temp.: temporal; Collab.: collaborative; BoW: Bag of Visual Words; Hierar.: hierarchical; Assign.: assignment; Decent.: decentralised; VPR: visual place recognition; TWs: tracked words.

Ref. Method	Approach	Collab.	Local desc.	Bin.	Global desc.	Tree	Online	Temp.
[33] DBoW	Tree of bin. words with hierar. clustering		BRIEF [17]	✓	BoW	✓		
[68] DBoW-O	Tree of bin. words with hierar. clustering		ORB [77]	✓	BoW	✓		
[36] iBoW-LCD	Adaptive tree of bin. words with hierar. clustering		ORB [77]	✓	BoW	✓	✓	
[79] HBST	Incremental and balanced bin. search tree		ORB [77]	✓		✓	✓	
[102] BoTW	Incremental list of TWs		SURF [12]				✓	✓
[21] DVPR	Decent. pre-defined assign. of visual words	✓	ORB [77]	✓	BoW	✓		
[19] DSLAM	Decent. pre-defined assign. of feature vector clusters	✓	ORB [77]	✓	NetVLAD [7]			
XC-PR	Decent. VPR with adaptive trees of stable TWs	✓	ORB [77]	✓		✓	✓	✓

in loop closure detection or Collaborative Visual SLAM.

To identify previously seen places, approaches for loop closure detection extract discriminative and compact representations of the images using *global* descriptors as the result of a direct transformation of the input images or the aggregation of *local* image features in a compact feature vector (e.g. BoW) [56]. Then, these approaches search a set of candidate places by matching global representations, followed by a validation step of the local features with a geometric model (*e.g.* epipolar constraint).

Examples of *direct global descriptors* are the handcrafted GIST [71] and histogram of gradients (HOG) [23], or the learning based NetVLAD [7] and DeepBit [53]. When using global descriptors, visual place recognition can also be seen as stand-alone image retrieval problem to support methods, such as deep learning frameworks, that can cope with illumination, appearance, seasonal, and day-night changes [7, 108]. NetVLAD is based on Convolutional Neural Networks and has been shown to be partially invariant to viewpoint and illumination changes, tolerant to partial occlusions, and to handle seasonal changes. NetVLAD aggregates the first order statistics of mid-level convolutional features, as residuals in different parts of the descriptors space weighted by a soft assignment, into a fixed-length representation. This aggregation can be used as last layer in any Convolutional Neural Network. DeepBit [53] also uses Convolutional Neural Networks to describe a whole image with a feature vector consisting of binary values for efficient and accurate image retrieval. While minimising the quantisation effects, DeepBit aims at preserving distinctiveness of the original image and being invariant to geometric variations.

BoW recursively quantises the descriptor space of *local features*, such as SIFT [55] or ORB [77], in a pre-defined number of clusters (visual words), for example using k-means, to form a (vocabulary) tree [33, 90, 70, 67]. Then, the histogram of visual word occurrences defines

the global descriptor. Exploiting the term frequency, indirect document frequency concept from information retrieval, BoW approaches adopt an indirect index to efficiently retrieve images associated to a word in the tree. While linear search and match over all possible features becomes expensive with a large number of features, the tree representation allows to efficiently match a query descriptor with a limited number of local features, *i.e.* those belonging to the corresponding word. To achieve real-time performance, DBoW [33] adopts binary features, coupled with the FAST detector, and keeps another index to relate images with the features in the visual words (direct index). To avoid consecutive images to compete with each other, DBoW [33] groups the retrieved images that are most similar to the current query image by matching BoW vectors. After verifying the temporal consistency of the current matched group with previous matches, DBoW selects and validates the image with the highest score by matching the binary features. Mur-Artal *et al.* [69] replace BRIEF features [17] with the rotation-invariant and scale-aware ORB features [77] to make DBoW more robust to geometric changes. For simplicity, we refer to this method as DBoW-O. Because of the oriented descriptor, DBoW-O adds a further orientation verification to make the descriptor matching faster and more robust. Exploiting 3D information, DBoW-O then uses similarity transformation, fusion of duplicate points, and graph optimisation to close the loop. However, to learn the vocabulary, BoW approaches [33, 69, 70] require a pre-training phase that may be time-consuming and dependent on the chosen dataset. This makes BoW approaches less adaptable to new and unknown scenes, requiring a priori knowledge of the environment. Moreover, more extensive computation would be necessary if considering to re-train the model, which cannot be feasible for on-the-fly applications.

To overcome this problem, other approaches learn the vocabulary while processing new frames of an image sequence (online) [102, 36, 79]. Inspired by the hierarchical structure proposed in Muja and Lowe's work [67] to index and match binary features, iBoW-LCD [36] uses an incremental and adaptive version of the tree based on an update policy. This update policy merges matching descriptors from the current image to the previous image, inserts new visual words if current descriptors are not matched with any previous descriptors, and removes temporarily new words that are not observed for a pre-defined number of times in a temporal window. iBoW-LCD [36] then dynamically groups past images with respect to the current query image by retaining only those images that are older than an adaptive window and whose score is higher than a pre-defined threshold. To increase efficiency, iBoW-LCD gives priority to a group

with lower score but that overlaps with the best group of the previous query, rather than estimating and selecting the best candidate. Instead of constructing a tree via hierarchical clustering, HBST [79] arranges binary features in the nodes of a binary search tree using as splitting criterion the binary value at indexes computed as optimal partitioning. This approximates the exhaustive search allowing efficient descriptor insertion and matching in logarithmic time. While binary features from new incoming images are accumulated within the binary search tree, HBST limits the growing depth of the tree by splitting a node into two leaves only if a maximum number of stored features is reached. While these approaches rely on a hierarchical structure to organise the local features, BoTW [102] forms an incremental list of *tracked words* that are obtained by tracking and averaging SURF features [12]. Also, each tracked word is paired with its length and the list of all the images where the local feature was observed (indirect index). BoTW then matches SURF features at the current frame against tracked words older than the most active tracked feature via a nearest neighbour strategy. A voting strategy selects the image with the majority of matched and visible tracked words as the best candidate to be validated for loop closure detection. As independent of the "term frequency" scoring technique, both HBST and BoTW can directly match local features using the indirect index. However, the main limitation of BoTW is that the matching of tracked words becomes computationally expensive over time. Indeed, the computational matching cost increases proportionally to the the number of tracked words inserted in the list, which grows linearly over time [79].

DSLAM [19], a decentralised and collaborative approach, assigns words of a pre-computed vocabulary to specific cameras (distributed inverted index). Given a query feature vector from a camera, the other cameras compute only a partial score as a response to the first camera. Then, the first camera determines which of the other cameras is the most likely to have seen a similar place. This approach scales similarly to a centralised method and avoids either to query all the other cameras or only cameras in a range [21]. While the authors initially relied on DBoW-O, they then showed that a direct global descriptor, such as NetVLAD [7], can be clustered as well and allows DSLAM to achieve higher performance [20]. However, binary features are still computed and matched across cameras for the geometric validation step. Moreover, not all binary features have a 3D correspondence in each camera, resulting in matches not useful or relevant for the global 3D reconstruction and/or relative localisation.

Table 2.2 summarises the most relevant works. For an in-depth analysis, we refer readers to

Lowry *et al.*'s survey [56].

Most of these loop closure detection strategies rely on vocabularies [19, 21, 33, 68, 69] or direct global descriptors [7, 20] that are trained offline, requiring expensive training phases, and are not easy to generalise to new and unknown scenes. Online approaches [36, 79, 102] can overcome these two limitations, but only BoTW [102] exploits temporal information to make the visual place recognition more robust to severe geometric differences. All the other approaches are based on image retrieval concepts with the goal of finding the most similar image despite appearance changes. However, the similarity considerably decreases when the same place of a scene is observed from different viewpoints, as in the case of collaborative systems. Even though temporal information is exploited, BoTW [102] lacks the efficiency of other approaches as visual words are organised in an incremental list instead of a tree.

2.5 Datasets

We discuss, in this section, datasets and benchmarks used in the literature for evaluating local image features.

The evaluation of local image features has been investigated, revisited and improved over years. Datasets for local image features can be categorised as planar versus non-planar based, and image versus patch based. Table 2.3 summarises the datasets, giving their relative properties and challenges.

Examples of image and planar based datasets are Oxford Affine Covariance Regions Dataset (ACRD) [61], Heinly's image matching [42], Fisher's synthetic [30], Viewpoints [65], Webcam [104], and EdgeFoci [120]. The ACRD dataset [61] consists of eight sequences of six images under five different conditions: image blur (*bikes* and *trees*), illumination changes (*leuven*), in-plane rotation changes and scale changes (*bark* and *boat*), viewpoint changes (*graf* and *wall*) and JPEG compression (*ubc*). Heinly *et al.* [42] extended ACRD with other five sequences under more severe transformations: pure rotation (*ceiling*, *rome* and *semper*), pure scaling (*venice*), and illumination changes (*day_night*). Figure 2.1 shows samples of sets from ACRD and Heinly's image matching. EdgeFoci [120] included five more sets that capture non-linear illumination variations and changes in background clutter to address more extreme cases, while Viewpoints [65] addresses the specific case of only extreme viewpoint variations with in-plane rotations up to 45 degrees, and Webcam [104] addresses extreme illumination, day/night or seasonal changes.

Table 2.3: Summary of existing image matching datasets. KEY – Ref.: reference. # Seq: number of sequences. # Img: number of images. Pl.: planar. N-pl.: non-planar. Illum.: illumination changes. Rot.: in-plane rotation changes. Viewp.: viewpoint. DayN: day/night changes. N-LN: non-linear changes. Seas.: seasonal changes.

Ref.	Dataset	# Seq	# Img	Categories				Transformations								
				Pl.	**N-Pl.**	**Img**	**Patch**	**Rot.**	**Scale**	**Viewp.**	**Illum.**	**DayN**	**Seas.**	**Blur**	**JPEG**	**N-LN**
[61]	ACRD	8	48	✓		✓		✓	✓	✓	✓			✓	✓	
[120]	EdgeFoci	5	38	✓		✓				✓	✓	✓				
[42]	Heinly's	5	42	✓	✓	✓		✓	✓			✓				
[30]	Synthetic	24	624	✓		✓		✓	✓	✓	✓			✓		✓
[104]	Webcam	6	120	✓		✓					✓	✓	✓			
[65]	Viewpoints	5	30	✓		✓		✓		✓						
[9]	HPatches	116	696	✓		✓	✓	✓	✓	✓	✓					
[1]	DTU	60	7140		✓	✓			✓	✓	✓					
[93]	Strecha's MVS	2	19		✓	✓				✓						
[111]	Phototourism	3	*		✓		✓		✓	✓	✓	✓				
[63]	PhotoSynth	30	7287		✓		✓		✓	✓	✓	✓	✓			
[84]	ETH Bench	19	∼16400	✓	✓	✓		✓	✓	✓	✓	✓		✓	✓	
	WISW	50	179	✓	✓		✓	✓	✓	✓	✓			✓		
	IMW	26	29796		✓	✓		✓	✓	✓	✓	✓	✓			

Unlike previous datasets, Synthetic [30] provided sets of images obtained by synthetic transformations applied to an original image for each set. However, the synthetic generation process does not model all of the noises that are captured by a real camera, making the dataset less challenging [9]. Figure 2.2 shows samples of sets from these additional image and planar based datasets.

Most of these images are acquired with respect to a planar scene and thus image pairs can be related by a homography transformation, or if images are acquired far away from the scene, the transformation between image pairs can be approximated to a homography. Because of this, Mikolajczyk *et al.* [61] proposed a semi-automatic method to compute and refine the homography between image pairs to use as ground-truth annotation along with the dataset. Therefore, ground-truth correspondences between interest points, detected for each image pair, can be found by transforming all the points from the reference image into the candidate image and, then, computing either the Euclidean distance in pixels (a maximum of 1.5 pixels was suggested in [61], while [42] used 2.5 pixels) or the amount of overlap between the ellipses, if the points were detected with an affine region detector [61] (a threshold of 50% is suggested).

Examples of image- and non-planar based datasets are the Multi-View Stereo (Strecha's MVS) [93] and DTU [1] datasets. Strecha's MVS [93] contains two sequences, *Fountain* (11 images) and *HerzJesu* (8 images), acquired with increasing variation of the viewpoint around the 3D scene. The dataset also provides 3D LIDAR-based geometry and camera poses. DTU [1] contains 60 sequences acquired in a controlled environment with a pre-defined lighting setting was

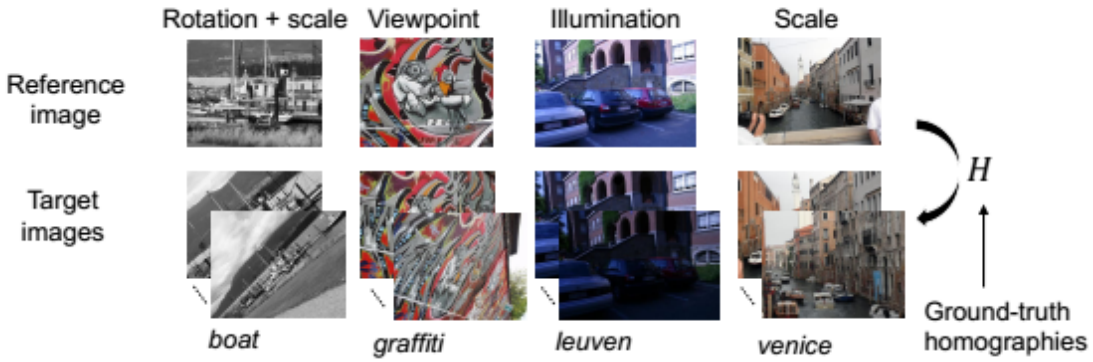


Figure 2.1: Sample of image and planar based sets from ACRD [61] (*boat*, *graffiti* and *leuven*) and Heinly's image matching [42] (*venice*) with corresponding challenges. Target images are related to reference images via homography transformations provided as ground-truth with the datasets.

Figure 2.2: Sample of images from planar datasets. From left to right: *Mexico* from Webcam [104], *Mario* from Viewpoints [65], *Small palace* from Synthetic [30], *Obama* from EdgeFoci [120], *London Bridge* and *Underground* from Hpatches [9]. For each original image on the top, we show two corresponding images under different transformations on the bottom.

used for 119 selected viewpoints with known camera poses. The sequences contain objects with varying material types and reflectance properties, and their 3D surface model is also provided. In addition to homography-based sets, Heinly's image matching [42] complements the Strecha's MVS with two additional sets of non-planar landmarks, such as *Reichstag*, and *Berliner-Dom*.

Examples of patch-based datasets are HPatches [9], Phototourism [111], and PhotoSynth [63] (see Figure 2.3). HPatches [9] contains 116 sequences of planar images - one reference image and five target images for each sequence - split into two groups: one whose main challenge is given by photometric changes, such as illumination variations; and the other whose main challenge is given by geometric transformations due to viewpoint changes. Some of the sequences are taken from existing datasets [1, 61]. Patches are extracted from reference images using common de-

Figure 2.3: Sample of images from non-planar datasets. From left to right: *Fountain* and *Herz-Jesu* from Strecha's MVS [93], *Reichstag* from Heinly's Image Matching [42], and *blankets* and *houses* from DTU [1]. For each set, images go under different geometric and photometric transformations, such as viewpoint and illumination changes (from top to bottom).

tectors [55, 60] and ground-truth correspondences are obtained by projecting the patches – after applying random transformations of three levels of increasing noise – from the reference image to the target images using the known homography transformations. The dataset aims to increase the diversity and the quantity of data as well as the reproducibility of the evaluation in three different tasks, such as patch verification, image matching and patch retrieval. Phototourism [111] contains a large number of annotated matching pairs of patches extracted from three photo collections of real world scenes (non-planar), namely *Liberty*, *Yosemite*, and *Notre-Dame*, using Structure from Motion. To increase the diversity in terms of scene content, illumination, and geometric variations, PhotoSynth [63] instead provides 30 scenes of 200 images on average and for each scene, patches and ground-truth correspondences are obtained through Structure from Motion (*e.g.* using COLMAP [83]).

Recently, several benchmarks have been proposed to evaluate local image features in large, diverse, and challenging conditions by exploiting existing datasets, introducing some new sequences, and proposing further performance measures. The ETH Local Features Benchmark [84][1] evaluates learned and histogram-based descriptors on the Heinly's image matching datasets and on several datasets for the 3D reconstruction task, included Strecha's MVS. The Which is which? evaluation benchmark (WISW)[2] provides patches extracted from 148 image pairs taken from existing datasets [1, 61, 65] or newly collected in both planar and non-planar scenes with cor-

[1] https://github.com/ahojnnes/local-feature-evaluation
[2] http://cvg.dsi.unifi.it/cvg/index.php?id=caip-2019-contest

responding evaluations. The Image Matching CVPR Workshop 2019 Challenge[3], instead, provides 26 photo-tourism image collections of popular landmarks from the Yahoo Flickr Creative Commons 100M (YFCC) dataset[4] and Reconstructing the world in six days [43]. The number of images varies for each set from 75 images to almost 4000. The dataset relates image pairs with ground-truth data, such as calibration data, camera poses and depth maps, obtained with a Structure-fro-Motion pipeline (*e.g.* COLMAP [84]). In addition to wide-baseline stereo matching, the dataset allows to evaluate local image feature in the task of 3D reconstruction.

2.6 Performance measures

The evaluation of local image features for image matching uses repeatability for assessing the detector performance, and mainly precision and recall as performance measures for the descriptor [61]. The *repeatability score* is the number of ground-truth correspondences over the number of features in common between an image pair. The number of features in common is given by the minimum between the number of features detected in the target image and the number of features that are transformed from the reference to the target image and lie within the target image boundaries. *Precision* (P) is the number of correct matches (or true positives, TP) over the total number of matches (true positives + false positives). *Recall* (R) is the number of correct matches over the number of ground-truth correspondences. From these measures, additional measures can be computed, such as the matching score and the putative match ratio [42], or the F_1 score. *Matching score* (MS) is the ratio between the correct matches and the number of features in common. The *putative match ratio* quantifies the selectivity of the local image descriptor as the ratio between the number of matches and the number of common features. F_1 score (or F_1 measure) is the harmonic mean between precision and recall: $F_1 = 2\frac{PR}{P+R}$.

Matching features are determined by the matching strategy, such as *threshold-based*, *nearest neighbour* and *nearest neighbour with Lowe's ratio test* [55, 61]. Threshold-based matching strategy determines the number of matches by setting a threshold on the descriptor distance and, therefore, a feature in the reference image can be matched with many features in the target image. The nearest neighbour matching strategy finds only one correspondence for each feature. If a feature is matched with multiple features in the second image, only the correspondence with the lowest distance is preserved as valid, while the other feature remains unmatched, unless a mutual

[3] https://image-matching-workshop.github.io/challenge/

[4] https://webscope.sandbox.yahoo.com/catalog.php?datatype=i&did=67

match is enforced. The Lowe's ratio test compares the distance of the first with the second best match and, if the ratio between the two distances is higher than a threshold, then the match is discarded. This condition attempts to remove possible ambiguities and false positives in the evaluation.

To compare methods independently of tuned parameters, or the threshold on the descriptor distance, precision-recall curves (*e.g.* the *recall vs 1-precision* curve [61]) can be generated by varying these parameters. The curve can be summarised in a single number by computing the Average Precision, *i.e.* the area under the curve, and then the methods are ranked accordingly [9, 26, 51, 116]. The *mean Average Precision* (mAP) instead aggregates the Average Precision results across multiple sets. Alternatively, for patch-based datasets, the relationship between precision and recall is captured by the Receiver Operating Characteristics curves. However, these curves are usually less representative for unbalanced data, and benchmarks prefer to rank methods using the false positive rate at 95% of true positive recall (FPR95) or the mAP [9, 111]. To better capture the overall performance, depending on the dataset, benchmarks can report multiple measures. However, there is no consistency in the chosen measures among the evaluations in the literature, leading to multiple inconsistencies when also ranking the methods. Therefore, recent benchmarks, such as HPatches [9] or the Image Matching CVPR Workshop 2019 Challenge, are attempting to overcome these issues by providing large datasets, multi-tasks, and corresponding performance measures to compare existing and new approaches. For example, IMW compares the performance of many local features when matching image pairs obtained from SfM reconstructions of selected landmarks (26), using matching score or mAP up to different thresholds on the estimated poses; or when performing Strucure from Motion from small subsets of the landmarks, using the previous, large reconstructions as reference annotation. Also HPatches adopts mAP as the main performance measure to rank and compare the methods, denoting the convergence of the community towards this metric as currently the most representative one.

2.7 Summary

In this chapter, we summarised and discussed existing local image features, spatio-temporal features, and methods to recognise images of previously seen places, usually adopted for loop closure detection in a single moving camera, but also applied to multiple moving cameras.

Local image features have been widely investigated and improved to achieve invariance to

different geometric transformations, robustness to photometric variations and image artefacts, and to be compact for efficient matching. These challenges prompted the design of methods that exploit local information, such image statistics and intensities (*e.g.* histogram-based descriptors), and more recent methods based on the learning from large datasets using CNNs (deep learning based features), either at the patch level or on the whole image along with the localisation of the features. Despite achievements and advancements in tasks, such as patch retrieval or patch verification, deep learning based features have not totally proved to outperform advanced handcrafted features, which are still widely used, for example, for 3D reconstruction (*e.g.* RootSIFT [8] or DSP-SIFT [26]) [84], or to be a practical alternative to the efficient binary features for real-time applications, such as navigation [19, 34, 68]. Moreover, the expensive training on existing datasets does not guarantee that deep learning features can generalise and adapt to new and unknown scenes, which can be crucial for applications in which people cannot intervene. Moreover, binary features were introduced to satisfy efficiency and compactness requirements, for example in object tracking or Simultaneous Localisation and Mapping, using supervised or unsupervised approaches to learn sampling patterns or projection functions offline. However, the matching accuracy and invariance to geometric differences are reduced for these binary features. Therefore, designing features that are both efficient and invariant to severe geometric differences is still very challenging.

We then reviewed local spatio-temporal features that can be extracted either in a fixed volume or online by tracking local image features. Temporal information could be exploited to increase the invariance to viewpoint variations; however, these features were mainly designed for in-camera tasks and, therefore, are unsuitable for matching across cameras with severe geometric differences. Moreover, most of the existing spatio-temporal features build on histogram-based features and cannot satisfy efficiency and compactness requirements when two moving cameras interact with each other.

As efficient and continuous matching can be achieved by aggregating local image features or using a global descriptor, we reviewed the task of visual place recognition, particularly when used within loop closure detection algorithms in a moving camera, and also extended to collaborative scenarios. While early methods relied on Bag of visual Words approaches, whose vocabulary tree was learned offline and cannot generalise to new scenes, most of the recent approaches build the tree online and incrementally, achieving higher performance. However, all these approaches

cluster features extracted for each image, and the temporal information is not considered, except for Bag of Tracked Words [102], which does not exploit the efficiency of the tree structure since the visual words are organised in a incremental list.

Finally, we discussed datasets, benchmarks, and performance measures for the evaluation of local image features. As spatio-temporal features are designed for in-camera tasks, such as human action recognition, the literature lacks datasets of multiple sequences recorded with independent moving cameras, and procedures to annotate and evaluate these features. For visual place recognition, existing datasets are proposed and annotated for either loop closure detection or image retrieval; however, there are no datasets with visual place recognition annotation across independent cameras that simultaneously move in the scene.

Chapter 3

Matching multi-scale and spatio-temporal features under geometric variations

3.1 Introduction

In this chapter, we investigate the challenging problem of view matching across independently moving cameras that observe the scene from different viewpoints and distances (see Figure 3.1). Local image features play an important role in matching images under different geometric and photometric transformations. However, the feature similarity normally decreases with the increase in viewpoint, scale, and illumination changes, also affecting considerably the matching performance. Moreover, features visible in one view may be occluded in another view, thus leading to matching ambiguities. In addition to this, when the camera is moving, efficiency should be taken into account for real-time applications or resource-constrained devices when extracting and matching local features. Therefore, we consider binary descriptors that can be encoded ten times faster than histogram-based features, and their representation are typically stored with only 32 bytes, whereas SIFT uses 128 bytes [13, 17, 51, 55, 77]. However, binary features are less robust to geometric variations than other features that are histogram-based [8, 12, 55, 61, 98, 106] or CNN-based [10, 27, 57, 62, 72, 96, 97].

In the first part of the chapter, we introduce novel descriptors to address the scale and viewpoint differences. We first address the scale difference problem and we propose MORB, a multi-scale binary descriptor that is based on ORB [77] and that can cope with large scale-variations between views to improve the accuracy of feature matching under scale changes. We then tackle

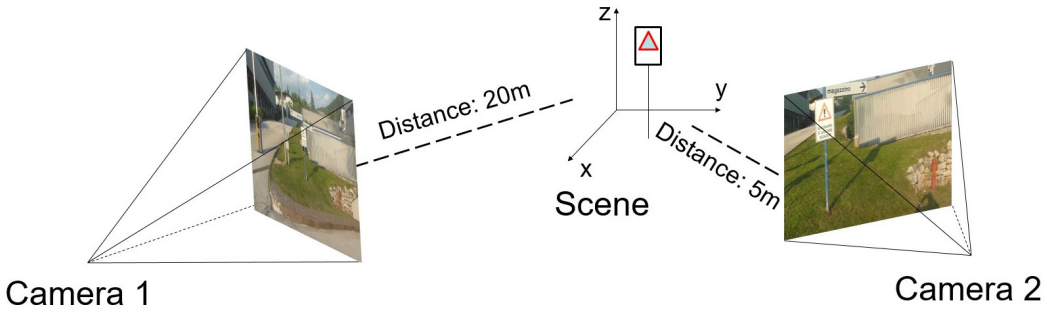


Figure 3.1: Example of two cameras observing a non-planar outdoor scene from different viewpoints and distances. The goal is to estimate independently for each camera a set of local features, either image-based or spatio-temporal, that can cope with the large geometric differences when matching the two views.

the viewpoint difference problem, and we propose to extract and match spatio-temporal descriptors for feature trajectories (or feature tracks) that capture the temporal changes of an interest point with uncalibrated and unsynchronised cameras. The proposed spatio-temporal features extract a sequence of ORB descriptors [77] and temporally pool the sequence to a compact fixed-length vector that encodes the most frequent values (dominant values) over time. We also extract a second vector that discriminates temporally unstable binary tests and acts as a selector of the first vector for feature matching. As last, we consider both scale and viewpoint changes in non-planar scenes and we propose a multi-scale temporal binary descriptor, named MST, that encodes the varying appearance of selected 3D points tracked by a moving camera.

In the second part of the chapter, we propose dissimilarity measures suitable for the proposed descriptors within a chosen matching strategy. For multi-scale descriptors, we propose the *scale-aware Hamming distance* to estimate the cross-scale match between MORB or MST descriptors across views to identify the best match and the scale difference of the features among images. We show that correct matches can be identified at descriptor scales that differ from the scale of the interest point. For the spatio-temporal descriptors, included MST, we propose a *selective weighted Hamming distance* to consider the unstability of the binary tests over time. For the multi-scale temporal descriptor, the two dissimilarity measures are combined together in a *scale-aware weighted Hamming distance*.

The rest of the chapter is organised as follows. Section 3.2 introduces the multi-scale binary descriptor. Section 3.3 describes the localisation, description and tracking of the spatio-temporal features and the temporal reduction for the proposed descriptor. Section 3.4 presents the proposed multi-scale temporal binary descriptor with its localisation, temporal reconstruction and

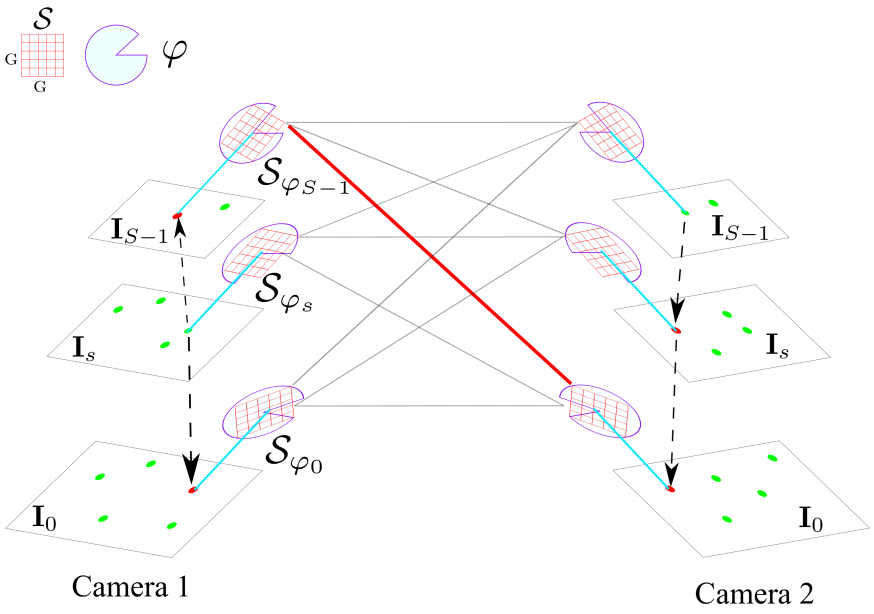


Figure 3.2: The multi-scale binary descriptor, MORB, and its cross-scale matching. Once an interest point is localised at a scale s (●), MORB samples its location (●) for each layer of an image pyramid, $\mathbf{I}_s$, and determines the patch orientation, φ_s. A descriptor based on a rotated sampling pattern for binary derivatives, $\mathcal{S}_{\varphi_s}$, is extracted for each scale s, keeping the $G \times G$ patch size fixed, and then contributes to the MORB descriptor. The matching across scales between MORBs from different viewpoints determines the scale difference (━ is the best match).

reduction. Section 3.5 describes the dissimilarity measures suitable for each descriptor and corresponding matching strategies. Section 3.6 discusses the proposed descriptors with respect to the state of the art. The evaluation of the proposed methods will be in Chapter 5.

3.2 Multi-scale binary descriptor

We propose MORB, a novel multi-scale binary descriptor that is based on ORB and that can cope with large scale variations between views. MORB describes an image patch at different scales using an oriented sampling pattern of intensity comparisons in a predefined set of pixel pairs. Descriptors extracted at different scales are appropriately rotated to adapt to the varying content within a patch and then concatenated to form a unique descriptor. How to match MORB descriptors across images using a cross-scale matching strategy based on a set-to-set minimum distance (*set2set mindist*) will be discussed in Section 3.5. Figure 3.2 illustrates the extraction of the MORB descriptor for interest points localised at different scales of image pyramids of two different images, as well as the cross-scale matching between these two descriptors, with the matched scales highlighted with a red line.

Let $\mathcal{I} = \{\mathbf{I}_s\}_{s=0}^{S-1}$ be a Gaussian pyramid of image $\mathbf{I}$, where each layer $\mathbf{I}_s$ is recursively down-

sampled by a factor λ, up to scale $S-1$. We apply in each $\mathbf{I}_s$ independently and adaptively the FAST detector [76] and retain only the F features across scales with the highest Harris[1] response [37]:

$$F_s = \begin{cases} \left\lceil \frac{1-\lambda^{-1}}{1-\lambda^{-S}} F \right\rceil & \text{if } s = 0 \\ \lceil \lambda^{-1} F_{s-1} \rceil & \text{if } 0 < s < S-1 \\ \max\left(F - \sum_{q=0}^{S-2} F_q, 0\right) & \text{if } s = S-1, \end{cases} \tag{3.1}$$

where the resulting coefficients sum to 1 and F_s is the number of features for each scale s.

After smoothing each layer $\mathbf{I}_s$ with a 2D Gaussian filter with size W and standard deviation σ, we extract the descriptor $\mathbf{d}_{f,s}$ using the rotated ORB sampling pattern on a $G \times G$ patch $\mathbf{p} = \psi(\mathbf{I}_s, \mathbf{x}_{f,s}, G)$ centred at each feature location:

$$\mathbf{d}_{f,s} = [\mathbf{d}_{f,s}(\mathbf{u}_1), ..., \mathbf{d}_{f,s}(\mathbf{u}_d), ..., \mathbf{d}_{f,s}(\mathbf{u}_D)], \tag{3.2}$$

where $\mathbf{u}_d = (\mathbf{u}_{d,1}, \mathbf{u}_{d,2})$ are the positions of each pixel pair defined by the sampling pattern $\mathcal{S}$, with $d = 1, ..., D$ (*e.g.* $D = 256$ for the ORB sampling pattern [77]), and $f = 1, \ldots, F$ is the index of the f-th feature.

The binary test on the intensity values $\mathbf{p}(\mathbf{u}_{d,1})$ and $\mathbf{p}(\mathbf{u}_{d,2})$ in patch $\mathbf{p}$, at scale s, of each pixel pair $\mathbf{u}_d$ of the sampling pattern is:

$$\mathbf{d}_{f,s}(\mathbf{u}_d) = \begin{cases} 1 & \text{if } \quad \mathbf{p}(\mathbf{u}_{d,1}) < \mathbf{p}(\mathbf{u}_{d,2}), \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

The pattern $\mathcal{S}$ consists of learnt pixel pairs with high variance and low correlation in their binary derivative [77]:

$$\mathcal{S} = \{\mathbf{u}_d = (\mathbf{u}_{d,1}, \mathbf{u}_{d,2})\}_{d=1}^{D}. \tag{3.4}$$

As scale variation is already contained in the Gaussian pyramid, we keep the patch size fixed across scales. This changes the portion of the scene captured by the patch at different scales. We also re-compute the orientation angle φ_s for each scale s. The angle φ_s is calculated with respect

[1] The Harris score is preferable to the FAST score as cornerness measure [77].

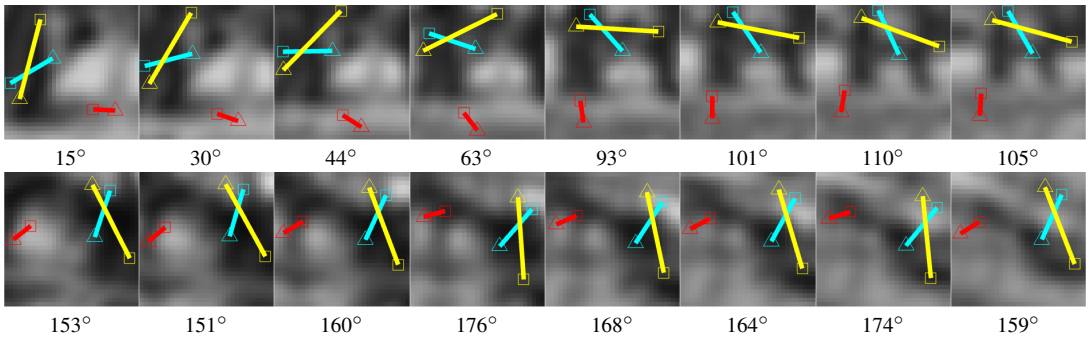


Figure 3.3: Sample patch orientation changes along the scales (from left to right) and across views (top row: view 1; bottom row: view 2) for the proposed MORB descriptor. For each patch, we show its orientation in degrees and 3 sample rods (red, cyan, yellow) from the ORB sampling pattern.

to the centre of mass of the patch defined by the intensity centroid [75]. Each $\mathbf{d}_{f,s}$ is extracted using the rotated pattern $\tilde{\mathcal{S}}_s$, after the rotation $\mathbf{R}_{\varphi_s} \in SO(2)$ is applied to each pixel in $\mathcal{S}$:

$$\tilde{\mathcal{S}}_s = \{(\mathbf{R}_{\varphi_s}\mathbf{u}_{d,1}, \mathbf{R}_{\varphi_s}\mathbf{u}_{d,2}) | (\mathbf{u}_{d,1}, \mathbf{u}_{d,2}) \in \mathcal{S}\}, \tag{3.5}$$

where $SO(2)$ is the special orthogonal group in $\mathbb{R}^2$ and consists of all orthogonal matrices of determinant 1 [58]. Figure 3.3 is an example of the rotated pattern at different scales.

The MORB descriptor $\mathbf{d}_f$ extracted at patch $\mathbf{p}$ of the f-th feature point concatenates the patch descriptors extracted at all layers of the image pyramid:

$$\mathbf{d}_f = [\mathbf{d}_{f,0}, \ldots, \mathbf{d}_{f,S-1}] \tag{3.6}$$

and can support feature matching across views with significant scale change.

However, to extract the multi-scale descriptor for each interest point, MORB scales its image coordinates for each layer s and approximates them by rounding. This can result in interest points whose distances to the image border at the coarsest scale of the Gaussian pyramid after scaling are smaller than half of the patch size G and thus inhibits the extraction of the multi-scale descriptor. We therefore discard these interest points that are too close to the borders. We also remove duplicates by discarding one interest point from every pair of interest points that are at most 2 pixels from each other when up-sampled to the original image scale.

3.3 Spatio-temporal binary descriptor

We propose the localisation and extraction of a novel single-scale spatio-temporal binary feature in an online way within image sequence as acquired by an uncalibrated moving camera. We accumulate temporal information by tracking local binary features, which encode intensity comparisons of pixel pairs in an image patch (*e.g.* ORB features [77]). We then encode the spatio-temporal features into fixed-length binary descriptors by selecting temporally dominant binary values. We then complement the descriptor with a binary vector that identifies intensity comparisons that are temporally stable.

3.3.1 Localisation and description

Let $\mathbf{I}_k$ be a (gray-scale) frame at time k captured by an uncalibrated and moving camera with unknown poses. We apply the FAST corner detector [76] in each $\mathbf{I}_k$ and retain the F features with the highest Harris response [37], which are at feature locations $\mathbf{x}_{1,k},\ldots,\mathbf{x}_{f,k},\ldots,\mathbf{x}_{F,k}$.

After smoothing $\mathbf{I}_k$ with a 2D Gaussian filter of size W and standard deviation σ, we extract a descriptor $\mathbf{d}_{f,k}$ for each feature location using the ORB [77] sampling pattern, $\mathcal{S}$, on a $G\times G$ patch $\mathbf{p}=\psi(\mathbf{I}_k,\mathbf{x}_{f,k},G)$ centred at each feature location $\mathbf{x}_{f,k}$, $\mathbf{d}_{f,k}$ (see Eq. 3.2 and Eq. 3.3, but using only the scale of the original image). To account for in-plane rotations, we compute the orientation angle $\varphi_{f,k}$ of the patch with respect to its centre of mass as defined by the intensity centroid method [75]. The descriptor $\mathbf{d}_{f,k}$ is then extracted, after applying the rotation $\mathbf{R}(\varphi_{f,k})\in SO(2)$ to the sampling pattern $\mathcal{S}$ using Eq. 3.5.

3.3.2 Tracking and reduction

We track the features between frame $\mathbf{I}_k$ and $\mathbf{I}_{k-1}$ by matching their descriptors with a nearest neighbour approach followed by a validation strategy to allow only one-to-one matches. For each feature from frame k we select the three closest features in frame $k-1$ by using as dissimilarity measure the Hamming distance: $\langle \mathbf{d}_{f,k}\oplus\mathbf{d}_{g,k-1},\mathbf{1}\rangle$, where $\oplus$ is the bit-wise XOR operator. Selecting the three closest features allows us to increase the number of matches and reduce the number of features unmatched, while avoiding addressing all possible combinations. After ranking all candidate matches according to their Hamming distances, we discard matches whose features in $\mathbf{I}_k$ are outside a gate of radius r of the features in $\mathbf{I}_{k-1}$. We also discard matches with a feature with higher similarity in another match.

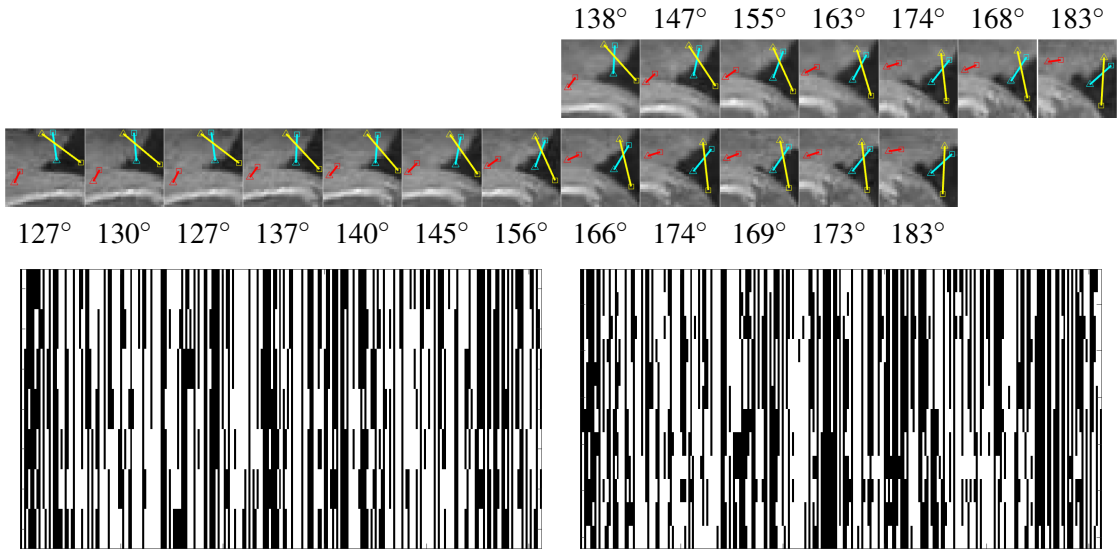


Figure 3.4: On top, sample patch orientation changes from frame 7 to frame 20 (from left to right) for the tracked ORB descriptor in one camera (first row) and the corresponding tracked ORB descriptor in an another camera (second row). For each patch, we show its orientation in degrees and 3 sample rods (red, cyan, yellow) from the ORB sampling pattern. At the bottom, the corresponding temporal ORB descriptors (differently from the patches, time is in a top-down representation), where we can see that some binary tests remain mostly stable on the vertical signals (black is a 0 and white is a 1).

The resulting trajectory, or feature track, of the binary feature i, localised in frame t_i and tracked over consecutive frames, until frame k_i is $\mathcal{T}_i = \{\mathbf{x}_{i,t_i}, \ldots, \mathbf{x}_{i,k_i}\}$ (see Figure 3.4). The length of $\mathcal{T}_i$ is $L_i = k_i - t_i + 1$. The spatio-temporal descriptor, $\mathcal{D}_i = \{\mathbf{d}_{i,t_i}, \ldots, \mathbf{d}_{i,k_i}\}$ is the set of descriptors accumulated over time and associated to $\mathcal{T}_i$.

We temporally reduce each $\mathcal{D}_i$ to a compact, fixed-length representation that captures the most frequent and the most stable binary values over time. We reduce $\mathcal{D}_i$ to a fixed-length vector $\mathbf{z}_i \in \{0,1\}^D$ by identifying the dominant (most frequent) binary values as

$$z_{i,d} = \begin{cases} 1 & \text{if} \quad \langle \mathbf{d}_{i,d}, \mathbf{1} \rangle > L_i/2, \\ 0 & \text{otherwise}, \end{cases} \tag{3.7}$$

where $\mathbf{d}_{i,d} = [\mathbf{d}_{i,t_i}(d), \ldots, \mathbf{d}_{i,k_i}(d)] \in \{0,1\}^{L_i}$ is the vector containing the temporal values of the element d, $\langle \cdot, \cdot \rangle$ is the (logical) dot product.

A binary test outcome should always output the same value, either 0 or 1, to be stable. However, to account for errors during the temporal matching due to photometric and/or geometric changes, we allow some variations in the binary test outcome, at a rate lower than 20% of the length of the feature track. We then compute a second set of descriptors that captures the temporal

changes, *i.e.* instability, of the binary tests in $\mathcal{D}_i$ via a bit-wise XOR ($\oplus$) of two consecutive binary descriptors: $\mathcal{D}'_i = \{\mathbf{d}_{i,k-1} \oplus \mathbf{d}_{i,k} \,|\, t_i < k \leq k_i\}$. Similar to $\mathbf{z}_i$, we reduce $\mathcal{D}'_i$ to $\mathbf{m}_i \in \{0,1\}^D$, the vector of the most stable binary values, as:

$$m_{i,d} = \begin{cases} 1 & \text{if} \quad \langle \mathbf{d}'_{i,d}, \mathbf{1} \rangle \leq 0.2(L_i - 1), \\ 0 & \text{otherwise.} \end{cases} \tag{3.8}$$

We refer to $\mathbf{z}_i$ and $\mathbf{m}_i$ as the vector of temporally dominant bits and vector of temporally stable bits, respectively. Therefore, $\mathbf{z}_i$ represents the Temporally Dominant (T-D) descriptor, while

$$\mathbf{w}_i = [\mathbf{z}_i, \mathbf{m}_i] \in \{0,1\}^{D \times 2} \tag{3.9}$$

is the Temporally Dominant-Stability (T-DS) descriptor.

3.4 Multi-scale temporal binary descriptor

We present a generic framework for binary descriptors that exploits the movement of a camera to selectively accumulate and encode temporal information about the appearance of a 3D point in a compact representation at multiple scales. To enable multi-scale extraction, unlike the previously proposed MORB descriptor that simply applies a cross-scale geometric verification to remove ambiguities, we design a feature suppression strategy that simultaneously enforces scale-invariance and favours a spatially uniform distribution during localisation. While the efficiency in the extraction of the descriptor at multiple scales decreases with the number of scales (a limitation common across SLS [39], DSP-SIFT [26], ASV [113], MORB (see Section 3.2), and our descriptor), using binary descriptors can mitigate this effect. Moreover, unlike the spatio-temporal binary descriptor, we use a pyramidal local search for feature point tracking [15] with respect to the feature point location at the highest resolution in the previous frame to increase the lifespan of feature tracks and to better capture appearance variations of the corresponding 3D points, and we also validate the feature tracks through 3D reconstruction.

3.4.1 Localisation

Let a local image feature represent the patch around image location $\mathbf{x} \in \mathbb{R}^2$ with a D-dimensional descriptor $\mathbf{d} \in \{0,1\}^D$. The number and spatial distribution of interest points over an image typ-

(a) (b)

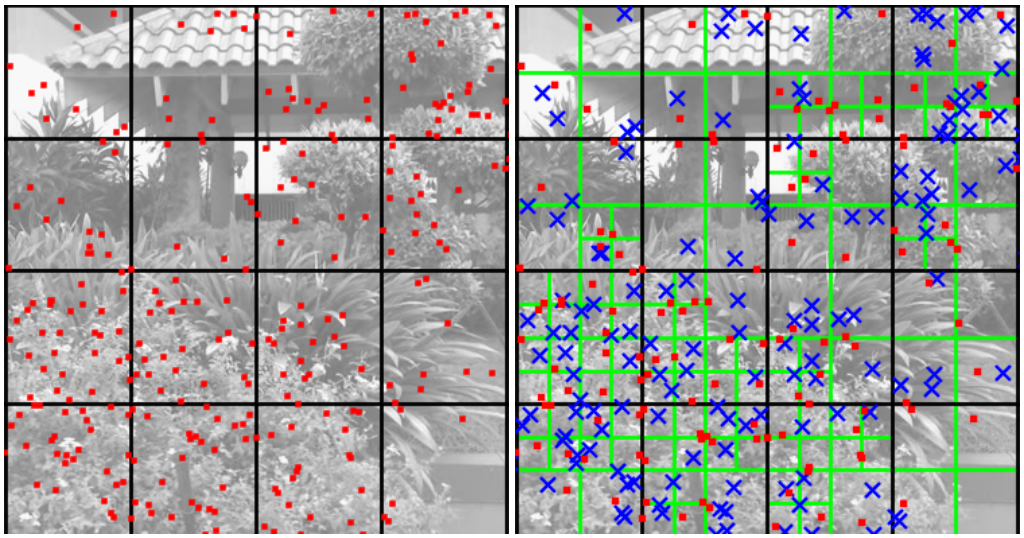
Figure 3.5: Sample results from two interest points suppression approaches. (a) Using only the cornerness response results in a few dense regions; whereas (b) using a regular grid and the cornerness response leads to a more uniform feature point distribution that is desirable when matching across very different viewpoints and scales. Legend: ● localised interest points and survived to the suppression approach.

ically depends on a decision on the corner response [76, 77]. However, using only the corner response can result in an undesirable concentration of interest points, thus reducing opportunities for matches from different viewpoints and scales (see Figure 3.5(a)). Moreover, when interest points are localised independently for each scale, redundancies can occur that generate ambiguities in the extracted descriptor [103]. To retain a maximum number of interest points without tuning the threshold of the corner response, we propose a suppression approach that simultaneously considers the corner response function to select the strongest points across nearby scales over a Gaussian pyramid (scale-invariance [50, 55]), and a regular grid to enforce uniformity in the interest point distribution over the image [68] (see Figure 3.5 (b)).

Let $\mathbf{I}_k$ be the frame at time k and $\mathcal{I}_k = \{\mathbf{I}_{s,k}\}_{s=0}^{S-1}$ be its (scale) pyramid [55, 77], where each layer $\mathbf{I}_{s,k}$ is recursively smoothed with a Gaussian convolutional kernel and down-sampled by a factor λ, up to scale $S-1$:

$$\mathbf{I}_{s,k}(\lambda^{-1}) = g(\mathbf{I}_{s-1,k}, \lambda^{-1}). \tag{3.10}$$

To allow the extraction of descriptors at multiple scales, we divide each $\mathbf{I}_{s,k}$ in a grid of $w \times w$ cells considering a scale-adaptive margin $B_s = \lambda^{S-s-1}G$ from the image borders, where $G \times G$ is the area around an interest point. We localise interest points with a good compromise between repeatability and extraction time [76, 77, 103]. Next, we suppress non-maxima points across scales by comparing the response with the eight neighbours at the same scale and with the nine neighbours in each of the nearest scales [50, 55].



(a) (b)

Figure 3.6: Feature suppression based on quadtree subdivision (blue cross: suppressed features). (a) Grid of cells superimposed on the image. (b) Cells with more than one interest point are split into four sub-cells (green blocks for each cell) and interest points (●) are accordingly assigned to each sub-cell. For each iteration, if the number of cells corresponds to the desired number of features, the interest points with higher Harris response [37] are retained in cells containing more than one interest point. Cells without points are not counted for the desired number of features.

As we obtain the Gaussian pyramid by using the terms of a geometric series as coefficients of proportionality based on the scale factor, λ, we proportionally distribute a number of localised interest points across scales, F_s, to determine the maximum number of features, F, using Eq. 3.1.

Therefore we retain only F_s interest points for each scale s in an iterative way [68]. For each iteration, we sort the cells based on their feature density in an ascending order (cells without points are not considered). We then subdivide the cells that contain more than one interest point into four sub-cells and interest points are assigned to each sub-cell based on their location. The iterative procedure ends when the number of (sub-)cells is equal or greater than F_s or all cells contain only one interest point. When a cell contains more than one interest point, we retain only the interest point with the highest corner response. Figure 3.6 illustrates the procedure for the retention of features based on their spatial distribution.

After localisation, we extract a descriptor for each interest point and then track the features. As our approach represents a 3D point associated to the trajectory of a feature, we will present our multi-scale spatio-temporal descriptor in Section 3.4.3 and we now focus on how to form a feature track and reconstruct its 3D point.

3.4.2 Temporal reconstruction

Once the feature points are localised and described, we estimate their trajectories over time. We use an iterative coarse-to-fine, local search by patch correlation through the scales of the image pyramid [15, 87]. While we observed that frame-to-frame matching, as used during the initialisation of ORB-SLAM [68], is subject to high-intermittent feature tracks[2], the pyramidal local search allows feature tracks to survive longer. We reduce the risks of early termination by comparing the descriptor of the candidate features at the current frame with a reference descriptor selected adaptively as the one with the shortest median distance from all the descriptors in the feature track. We thus terminate the trajectory if the distance of the descriptors is larger than a threshold γ, which represents the typical separation matching and non-matching feature distributions in the space of the Hamming distances, *e.g.* $\gamma = 50$ [17, 68]. As the camera moves, the number of visible features decreases over time and we detect new interest points every n frames over a masked version of the frame where all the pixels around the locations of existing trajectories are set to zero, and hence not considered. Then we initialise a new feature track for all the new interest points that are successfully tracked in the next frame.

Let us define the feature track as $\mathcal{T}_i = \{\mathbf{x}_{i,t_i}, \ldots, \mathbf{x}_{i,k_i}\}$, whose length is $L_i = k_i - t_i + 1$, where t_i and k_i are the indices of the first and last frame of the trajectory, respectively. Given the camera calibration information (*e.g.* obtained with the Zhang's method [119]), we derive from $\mathcal{T}_i$ the position of $\mathbf{X}_i$ by N-view triangulation with singular value decomposition [38]:

$$\mathbf{X}_i = \tau(\mathbf{x}_{i,t_i}, \ldots, \mathbf{x}_{i,k_i}, \mathbf{C}_{t_i}, \ldots, \mathbf{C}_{k_i}, \theta), \tag{3.11}$$

where $\tau(\cdot)$ is the triangulation function, $\mathbf{C}_{t_i}, \ldots, \mathbf{C}_{k_i}$ are the relative camera poses (*i.e.* position and orientation, which we assume to be available through an Inertial Measurement Unit, Odometry, or Structure from Motion), and θ contains the intrinsic camera parameters, such as focal length, principal point, and distortion coefficients.

To account for uncertainties in the feature point localisation, feature point tracking and triangulation steps, we validate the reconstructed 3D point with a maximum re-projection error of 5 pixels [68, 83] and by constraining the depth to be positive [38, 68].

Figure 3.7 illustrates the process of obtaining the binary spatio-temporal descriptor from a

[2]Frame-to-frame matching relies on the localisation strategy that selects different interest points for each frame and the matching is not constrained to a local area around each interest point in the previous frame.

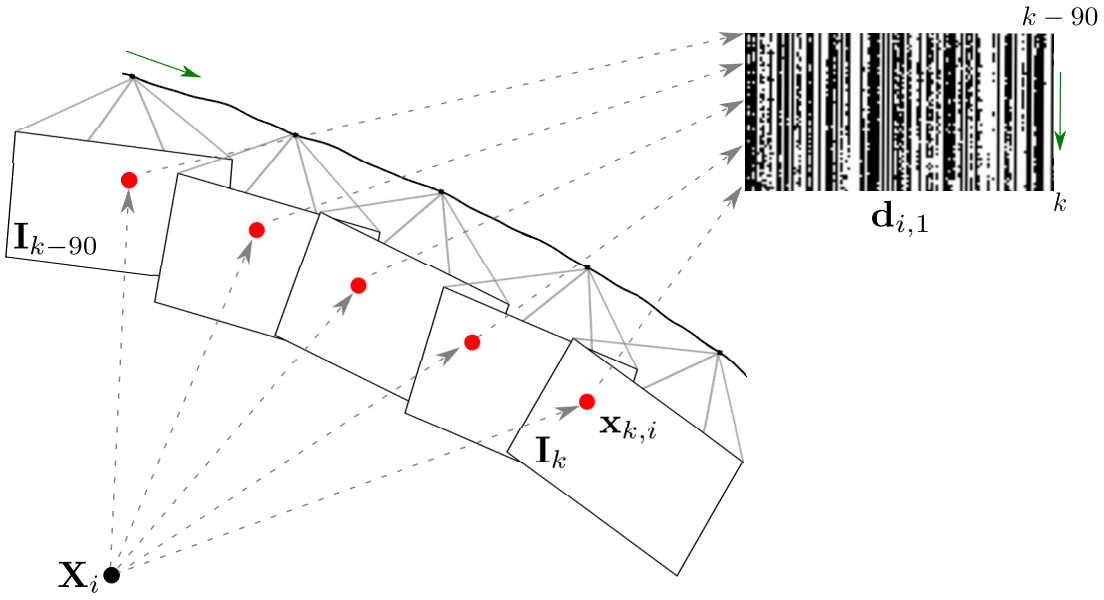


Figure 3.7: Illustration of the accumulation of binary descriptors of a tracked feature point $\mathbf{x}_{i,k}$ (●) representing a 3D point $\mathbf{X}_i$. The green arrow shows the direction of the time/trajectory.

feature point $\mathbf{x}_i$ tracked over consecutive frames ($\mathcal{T}_i$), and representing the corresponding 3D point, $\mathbf{X}_i$.

3.4.3 Multi-scale temporal descriptor

The feature tracking and associated 3D local reconstruction produce valid spatio-temporal features that we temporally reduce into a fixed-length descriptor considering the most frequent and stable binary tests.

We sample a patch around each point $\mathbf{x}_i$ of $\mathcal{T}_i$ at multiple scales with a pre-computed pattern $\mathcal{S}$ centred at $\mathbf{x}_{i,s,k}$ with $s = 1,\ldots,S$ and $k \in [t_i, k_i]$. To account for the rotation of the camera with respect to the 3D point, we rotate the patch towards the dominant orientation by $\varphi_{i,s,k}$ with respect to the centre of mass of the patch as defined by the intensity centroid [75]. We keep the size of the patch $G \times G$ fixed for each scale s of $\mathcal{I}_k$ and define the sampling pattern using Eq. 3.4. where $\mathbf{u}_{d,1}$ and $\mathbf{u}_{d,2}$ are pixel locations within the patch. After sampling using the rotated pattern:

$$\tilde{\mathcal{S}}_{i,s,k} = \{\mathbf{R}(\varphi_{i,s,k})\mathbf{u} : \mathbf{u} \in \mathcal{S}, \mathbf{R} \in SO(2)\}, \tag{3.12}$$

we generate the descriptor $\mathbf{d}_{i,s,k} \in \{0,1\}^D$, whose elements are resulted from the binary test between the pixel in pairs in $\tilde{\mathcal{S}}_{i,s,k}$ (see Eq. 3.3) [3, 17, 50, 77].

The descriptor $\mathcal{D}_i$ thus represents a set of patch descriptors at multiple scales and over time:

$$\mathcal{D}_i = \{\mathbf{d}_{i,0,t_i}, \ldots, \mathbf{d}_{i,S-1,t_i}, \ldots, \mathbf{d}_{i,0,k_i}, \ldots, \mathbf{d}_{i,S-1,k_i}\}. \tag{3.13}$$

We propose to represent the interest point with a more compact and fixed-length representation that captures the most representative tests of each 3D point as seen by a calibrated moving camera (see Figure 3.8).

For each scale s, we reduce the subset $\mathcal{D}_{i,s}$ to a fixed-length vector $\mathbf{z}_{i,s} \in \{0,1\}^D$ by accumulating the binary test values over time and identifying the dominant binary value using Eq. 3.7; and we then compute a second set, $\mathcal{D}'_{i,s} \in \{0,1\}^{(L_i-1)\times D}$, that captures the temporal changes, *i.e.* instability, of the binary tests in $\mathcal{D}_{i,s}$ via a bit-wise XOR of two consecutive binary descriptors using Eq. 3.8.

The dimensionality of the MST descriptor, $2 \times D \times S$, depends on the chosen number of scales, S, the length of the vector of temporally dominant bits and the vector of temporally stable bits, $2 \times D$. Note that D depends on the dimensionality of the specific employed image-based binary descriptor. Moreover, the total number of binary tests performed by MST depends on the length, L_i, of the feature trajectory. For example, considering 5 scales, a binary descriptor such as ORB ($D = 256$), and a maximum length of 50 frames, the minimum number of binary tests is 6400 (the maximum is 64000), and the dimensionality of MST is 2560 bits.

3.5 Descriptor matching

We aim to find a set of matches across cameras using the most suitable dissimilarity measures for each proposed descriptor.

As the scale at which multi-scale features, such as MORB and MST, should be matched is unknown, we cannot directly apply for matching nearest neighbour [61] or bag of words [33]. For this reason, we propose to estimate the minimum cross-scale distance between feature pairs (*scale-aware Hamming distance*). We discuss and validate alternative but worse performing scale-aware matching strategies in Appendix B. For the spatio-temporal features, including MST, we propose a *selective weighted Hamming distance* that uses the additional vector to ignore the corresponding binary values in the fixed-length binary descriptor when matching the features across cameras.

The set of putative matches is therefore determined by estimating the dissimilarity measure

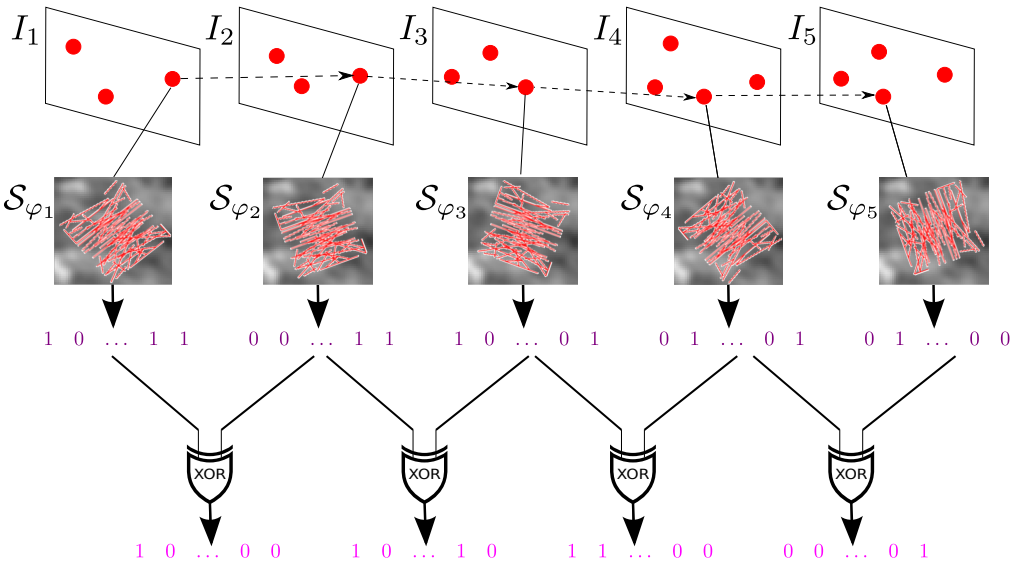


Figure 3.8: Extraction of the temporal binary descriptor at a single scale. The location of the interest point in the first frame is tracked in successive frames. For each frame the rotated sampling pattern is extracted forming a set of binary vectors (purple). We then compute the derivative (XOR operation) between consecutive binary vectors to estimate a second set of binary vectors (magenta) containing the frame-to-frame stability. For each set, we sum the vectors followed by threshold to obtain a vector of the most frequent binary values and a vector of the most stable tests over time, respectively.

between feature pairs within a similarity matching strategy such as threshold-based or nearest neighbour [61]. The ratio test between the distance of the first and second nearest neighbours can also be computed to remove possible ambiguities [55].

We now introduce the dissimilarity measure for each descriptor and the associated matching strategy.

3.5.1 Scale-aware Hamming distance

Let $\mathbf{d}_f$ and $\mathbf{d}_g$ be the multi-scale descriptors of an interest point f in one view and an interest point g in another view, respectively.

We first compute an all-to-all single descriptor distance across scales between each $\mathbf{d}_{f,s}$ and $\mathbf{d}_{f,l}$, and then we take the minimum of the computed distances as the cross-scale distance between the interest points:

$$h_{f,g}(s^*, l^*) = \min_{s,l} \langle \mathbf{d}_{f,s} \oplus \mathbf{d}_{f,l}, \mathbf{1} \rangle, \tag{3.14}$$

where $\oplus$ is the XOR operator and $\langle \mathbf{d}_{f,s} \oplus \mathbf{d}_{f,l}, \mathbf{1} \rangle$ is the Hamming distance between two single descriptors across scales. The scales where the minimum match is found, s^* and l^*, determine

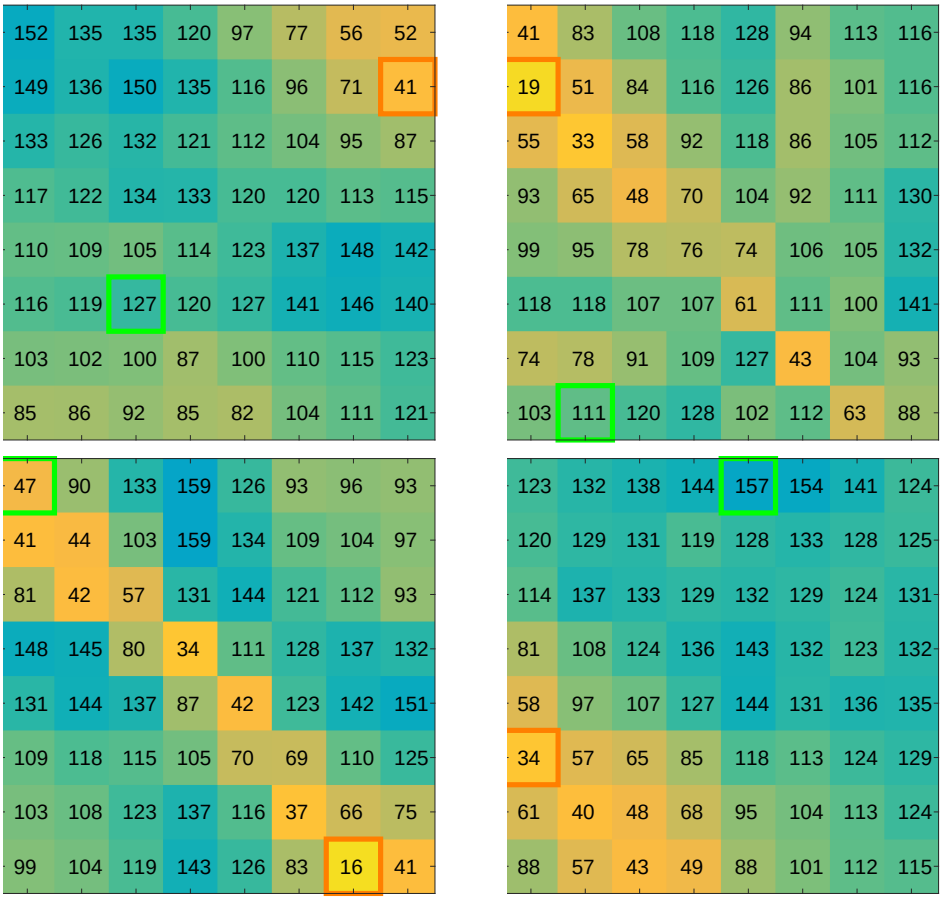


Figure 3.9: Hamming distance matrices across scales ($S = 8$) for a sample of four pairs of MORB descriptors. Green boxes denote the scales where interest points are localised. Orange boxes denote the scales of the minimum Hamming distance (of correct multi-view matches). Note the difference between the scales where the interest points are localised and where the descriptors are matched. Scales for the MORB descriptor from one view are on the y-axis (top-to-bottom), while scales for the MORB descriptor from another view are on the x-axis (left-to-right).

the scale offset between the two interest points ($|s^* - l^*|$).

Figure 3.9 shows examples of four cross-scale Hamming distance matrices between matched MORB descriptors. Figure 3.10 shows an example of cross-scale matching, where the match occurs at scales that are different from the localisation scales.

The set of putative matches $\mathcal{V}$ is estimated using nearest neighbour [61] and with a threshold γ on the descriptor distance to separate true positive and false positive putative matches. While the distribution of false positives can lie on high descriptor distances, the distribution of correct matches covers the low ones [17]. We obtain a set of matches between two views as

$$\mathcal{N} = \left\{ (f^*, g) \mid f^* = \underset{f \in \mathcal{F}}{\arg\min}\, h_{f,g}, g \in \mathcal{F}, h_{f,g} \leq \gamma \right\}, \tag{3.15}$$

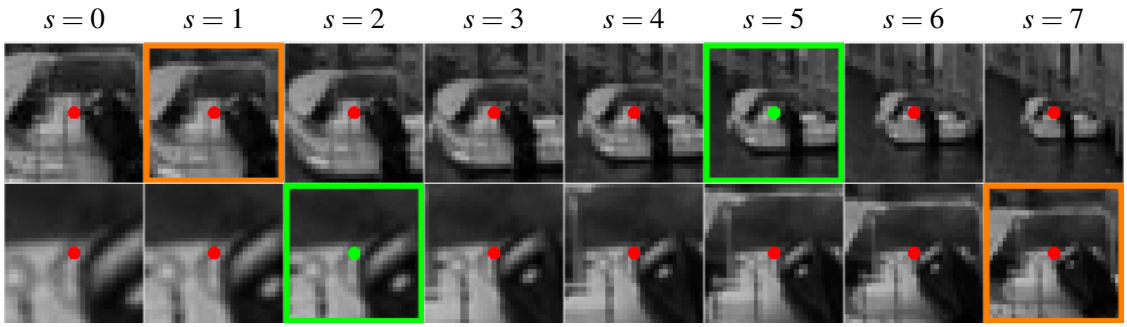


Figure 3.10: Sample corresponding patches at multiple scales ($s = 0,\ldots,7$) across views with considerable scale variation (top row: view 1; bottom row: view 2). Note the difference between the scales where the interest points are localised (green squares) and where the MORB descriptors are matched (orange squares). This case is related to the top-left matrix in Figure 3.9.

where $\mathcal{F} = \{1,\ldots,F\}$. Similarly, we obtain the set of reverse matches as

$$\mathcal{M} = \left\{ (f, g^*) \mid g^* = \arg\min_{g \in \mathcal{F}} h_{f,g}, f \in \mathcal{F}, h_{f,g} \leq \gamma \right\}. \tag{3.16}$$

The set of valid matches is then $\mathcal{V} = \mathcal{N} \cap \mathcal{M}$. We analyse the impact of the threshold on the effectiveness of our approach in Chapter 5.

3.5.2 Selective weighted Hamming distance

Let i be the index of a T-DS descriptor $\mathbf{w}_i$ in one view and q the index of a query T-DS descriptor $\mathbf{w}_q$ in another view.

We first remove temporally unstable bits of $\mathbf{z}_i$ and $\mathbf{z}_q$ (see Figure 3.11) by applying in turn the additional descriptors $\mathbf{m}_i$ and $\mathbf{m}_q$ to the XOR operation between $\mathbf{z}_i$ and $\mathbf{z}_q$ through the weighted Hamming distance [11]. Let the masked Hamming distance using only $\mathbf{m}_i$ be defined as $\langle \mathbf{m}_i, \mathbf{z}_i \oplus \mathbf{z}_q \rangle$, where $\oplus$ is the XOR operator. Let the number of stable binary tests for $\mathbf{z}_i$ be defined as $M_i = \langle \mathbf{m}_i, \mathbf{1} \rangle$ and, similarly, M_q for $\mathbf{z}_q$.

We then compute the final dissimilarity measure between two descriptors as a weighted linear combination of two masked Hamming distances:

$$h_{i,q} = \frac{M_i \langle \mathbf{m}_i, \mathbf{z}_i \oplus \mathbf{z}_q \rangle + M_q \langle \mathbf{m}_q, \mathbf{z}_i \oplus \mathbf{z}_q \rangle}{M_i + M_q}. \tag{3.17}$$

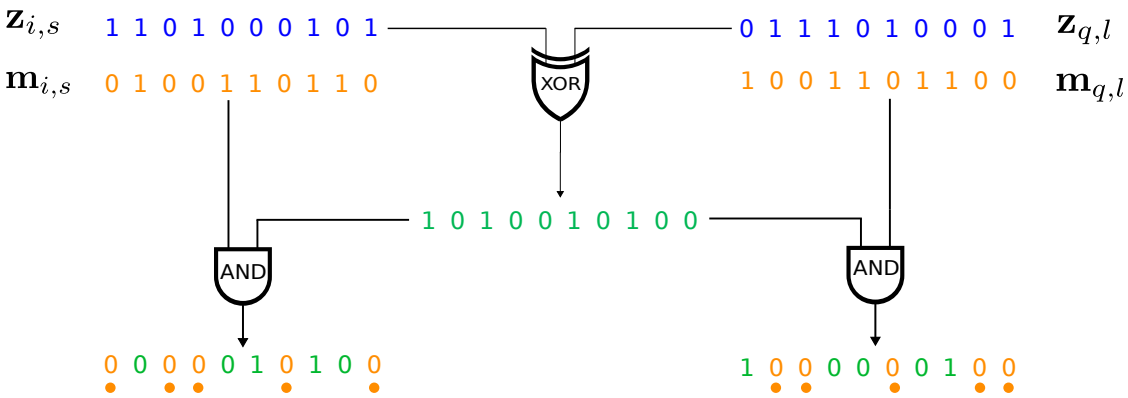


Figure 3.11: Graphical representation of the descriptor matching using the temporally dominant and temporally stable vectors at a single scale between descriptors of two different cameras. After selecting the stable bits of the resulting difference vector between the temporally dominant vectors, the weighted Hamming distance is applied (see Eq.3.17). Legend: ● unstable bits.

3.5.3 Scale-aware weighted Hamming distance

To handle the unknown scale difference between multi-scale temporal descriptors, we use the cross-scale matching strategy of MORB with the selective weighted Hamming distance of the spatio-temporal descriptor.

For each pair of MST descriptors, we first compute the selective weighted Hamming distance across scales, $h_{i,q}(s,l)$, as:

$$h_{i,q}(s,l) = \frac{M_{i,s}\langle \mathbf{m}_{i,s}, \mathbf{z}_{i,s} \oplus \mathbf{z}_{q,l} \rangle + M_{q,l}\langle \mathbf{m}_{q,l}, \mathbf{z}_{i,s} \oplus \mathbf{z}_{q,l} \rangle}{M_{i,s} + M_{q,l}}, \tag{3.18}$$

and we then identify the minimum across scales as

$$h_{i,q}(s^*,l^*) = \min_{s,l} h_{i,q}(s,l). \tag{3.19}$$

The scales where the minimum match is found, s^* and l^*, determine the scale offset between the two interest points ($|s^* - l^*|$).

To remove possible ambiguities, we determine the final set of matches through nearest neighbour followed by the Lowe's ratio test that validates a match only if the similarity distance of the closest neighbour is smaller than the distance from the second nearest neighbour [55, 61].

3.6 Discussion

In this chapter, we proposed novel descriptors and associated dissimilarity measures for matching features between cameras with different viewpoint and scale variations.

To handle scale variations between images, state-of-the-art approaches localise and describe interest points at multiple scales of an image pyramid. However, descriptors extracted *at the scale* where the interest point is localised [50, 55, 77] can be inaccurate when matching across images with severe scale variations [113]. Moreover, redundancies and ambiguities may arise if interest points are localised independently for each scale (*e.g.* ORB [77]), and can be avoided by suppressing non-maxima across scales [103] (*e.g.* SIFT [54] or BRISK [50]). Descriptors can also be extracted at *multiple scales* of a Gaussian pyramid to capture multi-scale information of an interest point [26, 39, 113]. Coarser levels allow one to distinguish locally repeated patterns, whereas finer levels capture subtle changes thus helping to discriminate nearby points [64].

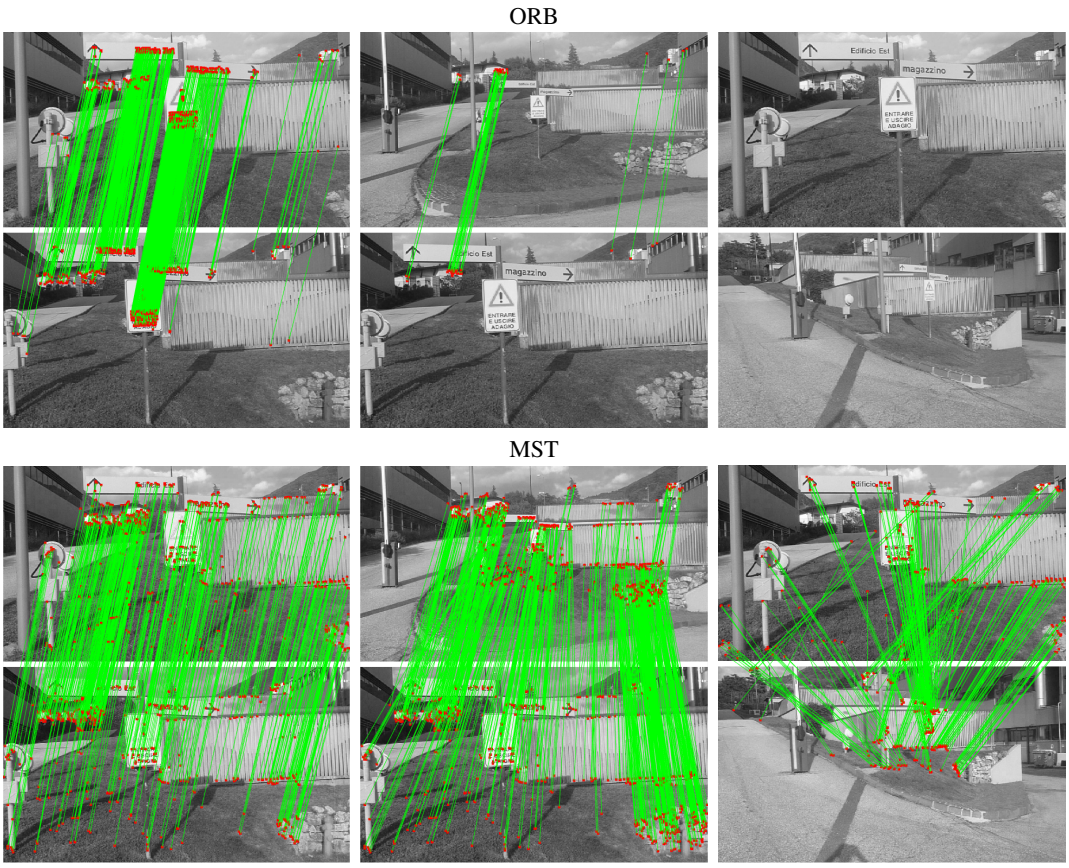
While histogram-based features can reduce the multi-scale extraction to a compact fixed-length descriptor, the extraction time depends on the chosen underlying feature and inevitably becomes proportional to the number of scales. We therefore investigated and designed a novel generic framework to extract a multi-scale descriptor using the efficient binary features, such as ORB. The proposed multi-scale binary descriptor, MORB, is coupled with a scale-aware nearest neighbour matching strategy that estimates the minimum cross-scale distance between two MORB descriptors and, as a by-product, can infer the scale offset between pairs of local features. The matched scales tend to differ from the scales where the interest points were localised. However, unlike histogram-based approaches, it is not straightforward to reduce the multi-scale binary descriptor to a compact representation and the bottleneck of the computational time is moved to the matching phase. Nevertheless, all the multi-scale descriptors are inadequate under severe viewpoint changes.

To handle this problem, we then investigated the problem of matching spatio-temporal features extracted from image sequences acquired by independently moving cameras. We proposed a spatio-temporal descriptor obtained by tracking and accumulating binary features [77]. As matching the high-dimensional descriptors is computationally expensive, we reduced the set of descriptors associated to a feature track into a fixed-length binary descriptor by selecting the temporally dominant values. We also complemented this descriptor with an additional vector that encodes the temporal stability of each binary test and ignores those binary values in the first

descriptor using a selective weighted Hamming distance when matching features.

To handle both scale and viewpoint differences, we proposed MST, a novel multi-scale temporal descriptor that captures appearance variations of a 3D scene point as observed by a moving camera. This compact descriptor selectively encodes the temporal information associated with the 3D point to improve robustness to view differences. In particular, as for the spatio-temporal features, we proposed a temporal reduction approach to encode the most frequent and stable binary values, so that the descriptor identifies temporally dominant values and the most stable tests over time. Moreover, to handle scale variations, the proposed descriptor relies on a multi-scale feature extraction and representation associated with a cross-scale matching strategy using the selective weighted Hamming distance as the dissimilarity measure. We will show in Chapter 5 that the proposed descriptor is generic for a range of binary descriptors. We show in Figure 3.12 a qualitative example of matching results with MST compared to ORB features [77] on a sample of image pairs with a similar viewpoint, different scale, and different viewpoint from the *gate* scenario used in Chapter 5. Note that the number of ORB matches can be even lower if an interest point is associated with a reconstructed 3D point as our approach does.

As our approach encodes the temporal information of feature tracks in a compact descriptor, MST differs from Daisy-3D [99], which concatenates tracked 2D Daisy features in a fixed window thus limiting the amount of information and variations captured by the spatio-temporal feature and requiring an expensive matching approach between cameras. MST also differs from LMED [68], which uses the ORB binary descriptor and selects the single descriptor over time with the least median distance with respect to all the tracked ORB descriptors within the feature track. While the chosen descriptor can reduce drifts in the feature tracks, this descriptor may not be suitable when matching features across cameras. Unlike STB [52], which describes the trajectory and temporal gradients of a fixed-size spatio-temporal volume, we obtain varying-length spatio-temporal features by directly accumulating image-based binary features, followed by a reduction to a compact, fixed-size representation. Moreover, MST handles scale differences when matching different views through multi-scale extraction and representation. Finally, unlike BOLD [11], which computes the stability vector with small geometric variations of the sampling pattern, we determine the stability by exploiting the temporal variations within a feature track. The stability is thus used as a selector when computing the distance between MST descriptors.



(a) Similar (b) Scale difference (c) Viewpoint difference

Figure 3.12: Matching performance of MST versus ORB [77] on *gate* (an outdoor scenario of collected sequences, see Appendix A). Green lines denote correct matches. When increasing the difference in scale (b) and viewpoint (c), the performance of ORB considerably decreases, while MST can handle the geometric variation. ORB features are matched using the nearest neighbour strategy with the distance ratio test [55], and a descriptor distance threshold. MST matches are based on the re-projection of the reconstructed 3D points in the selected frames and those that contains occluded points on the image are manually removed.

Chapter 4

Cross-camera place recognition

4.1 Introduction

In this chapter, we investigate the problem of view matching across two uncalibrated cameras that independently move in an unknown environment to identify previously seen places over time. While we do not make any assumption on the motion of the cameras, the problem requires us to design an approach that selects distinctive and compact descriptors from features observed in multiple frames, effectively identifies informative features to share across cameras, and recognises previously seen places within a camera while efficiently matching query descriptors received from another camera.

Recognising previously seen places is usually addressed within a single moving camera to support the self-correction of camera ego-motion estimations for reducing temporally accumulated drifts and reconstruction inconsistencies through loop closure detection in Visual SLAM [33, 34, 36, 68, 69, 102]. However, loop closure detection constrains viewpoint differences to be small ($< 30°$) [79], by assuming the platform (*e.g.* an autonomous vehicle) to move in a structured environment (*e.g.* a road), and observing previously seen areas from the same direction.

For other applications, such as collaborative augmented reality and gaming, people freely move in the environment wearing or holding cameras that observe portions of the scene from different angles and positions (viewpoint difference) and at different distances (scale difference). However, existing approaches extend monocular SLAM (*e.g.* PTAM [46], ORB-SLAM [68]) and

loop closure detection to operate across cameras for collaborative SLAM [19, 32, 74, 82, 121]. CoSLAM [121] and C^{2}TAM [74] extend PTAM and use specific mechanisms to recognise similar places, as PTAM does not use loop closure detection. CoSLAM estimates the visual overlap between the area spanned by a minimum number of 3D points projected in the view of another camera and the image area, and requires a large visual overlap ($> 70\%$). C^{2}TAM, instead, matches global descriptors obtained as a down-sampled and filtered version of the original image, followed by a search of correspondences between projected 3D points. CCM-SLAM [82] and DSLAM [19] extend ORB-SLAM and uses its BoW-based loop closure detection [69] to retrieve similar places across local maps of reconstructed 3D points. CoSLAM, C^{2}TAM, and CCM-SLAM are centralised approaches that perform map merging for global and consistent 3D reconstruction, whereas DSLAM decentralises both VPR and relative camera-pose estimation. Approaches such as CCM-SLAM [82] and DSLAM [19] adopt binary features for their compactness and efficiency in extraction and matching [17, 77]; however, binary features are less robust to large perspective differences [11]. Moreover, these approaches enforce high similarity between images when matching global descriptors due to the limited invariance to increasing viewpoint differences [7, 53], and validate only places whose binary features are associated to reconstructed 3D points [19, 68, 82]. Therefore, this considerably reduces the number of usable binary features.

To address these limitations, we present in this chapter XC-PR, a novel Cross-Camera Place Recognition approach that selects distinctive descriptors from binary features observed in multiple frames and effectively identifies informative features to share across cameras. XC-PR improves the matching accuracy by forming, for each camera independently, stable tracked words (TWs) that are obtained by associating binary features and temporally compressing their accumulated descriptors to a fixed-length representation. This representation preserves the most persistent values, which are more robust to temporal changes occurring while a local feature is tracked. This formulation of the TWs was previously presented in Chapter 3 and we briefly review it in Section 4.2 with some additional details for XC-PR.

As the number of TWs grows over time, matching them across cameras with a linear search may become computationally intractable. To enable efficient searching and matching, we insert TWs from automatically selected frames into a hierarchical structure, an Adaptive Tree of Stable Tracked Words (ATST). In Section 4.3, we formulate ATST as a search tree that adapts

over time through the insertion of new TWs and the removal of short TWs. Unlike BoW-based approaches [33, 70], ATST operates directly on the original descriptors instead of counting the frequency of quantised descriptor clusters that original descriptors may be associated with. When the number of tracked binary features is reduced due to the view change caused by the camera motion, a camera localises new binary features and updates ATST. In Section 4.4, we then present how the camera then shares a subset of TWs along with the image coordinates selected from the frame with the largest number of corresponding binary features, within an adaptive temporal window. XC-PR finally recognises a place within the camera that receives the query TWs by identifying and geometrically validating a previous frame with the largest number of matched TWs. The evaluation of the proposed XC-PR will be in Chapter 5.

4.2 Computing stable tracked words

Let a local feature comprise an image location (an interest point) $\mathbf{x} \in \mathbb{R}^2$ and a D-dimensional descriptor $\mathbf{d} \in \{0,1\}^D$ of the patch around $\mathbf{x}$. The descriptor may encode the result of binary tests (intensity comparisons) of pixel pairs within the patch [17, 77]. We refer to a local feature as binary feature, as its descriptor consists of binary values.

Let $\mathcal{T}_i = \{\mathbf{x}_{i,t_i}, \ldots, \mathbf{x}_{i,k_i}\}$ be the trajectory, or feature track, of binary feature i, localised in frame t_i and tracked over consecutive frames, until frame k_i. The length of $\mathcal{T}_i$ is $L_i = k_i - t_i + 1$. The larger L_i, the more comprehensive the description of the surrounding of the corresponding 3D point, and therefore the more likely a matching is from another view, in the absence of occlusions. We compute each feature track $\mathcal{T}_i$ with an iterative coarse-to-fine, local search by patch correlation to handle large camera motions between consecutive frames [15]. Despite surviving longer, this frame-to-frame tracking strategy may result in a feature track drifting towards a different implicit 3D point. Alternatively, tracking can be performed between the current frame and a reference frame, *e.g.* the first frame where the feature is localised, but in this case the feature track may terminate too early due to changes in the appearance of neighbourhood pixels. Therefore, we choose to adopt a hybrid option between the two techniques.

For each frame k, we compute the descriptor $\mathbf{d}_{i,k}$ corresponding to the tracked location $\mathbf{x}_{i,k}$ and we compare $\mathbf{d}_{i,k}$ with a reference descriptor adaptively selected within $\mathcal{T}_i$ to reduce the risk of early termination [68]. The reference descriptor is selected as the one with the shortest median distance from all the descriptors currently in the feature track, except the current descriptor. If

the distance between the current and reference descriptors is larger than a pre-defined threshold γ, then the trajectory terminates. This threshold is selected in such a way that matching and non-matching feature distributions are separated in the space of the Hamming distances [17].

Figure 4.1 shows the distribution of the TWs in the sequences of two testing scenarios with respect to their length. The distribution decreases exponentially with most of the TWs having a length between 10 and 50, and a few surviving a large number of frames. Given the observed distribution, feature trajectories may be terminated around 50 frames to avoid drifts of the tracked interest point with respect to the implicit associated point in the 3D world[1].

Let the set of descriptors accumulated over time and associated to $\mathcal{T}_i$ be $\mathcal{D}_i = \{\mathbf{d}_{i,t_i}, \ldots, \mathbf{d}_{i,k_i}\}$. We temporally reduce each $\mathcal{D}_i$ to a compact, fixed-length representation, $\mathbf{w}_i = [\mathbf{z}_i, \mathbf{m}_i] \in \{0,1\}^{D\times 2}$ (stable TW), that captures the most frequent and the most stable binary values over time (see Eq. 3.9). We reduce $\mathcal{D}_i$ to a fixed-length vector $\mathbf{z}_i \in \{0,1\}^D$ by identifying the most frequent binary values using Eq. 3.8. We then compute a second set of descriptors to capture the temporal changes, *i.e.* instability, of the binary tests in $\mathcal{D}_i$ using a bit-wise XOR ($\oplus$) of two consecutive binary descriptors, $\mathcal{D}'_i = \{\mathbf{d}_{i,k-1} \oplus \mathbf{d}_{i,k} \,|\, t_i < k \leq k_i\}$, and we reduce $\mathcal{D}'_i$ to $\mathbf{m}_i \in \{0,1\}^D$, the vector of the most stable binary values, using Eq. 3.8. Note that $\mathbf{m}_i$ may degenerate into all zero values, *e.g.* using frame-to-frame tracking. However, we show in Figure 4.1 that our hybrid tracking strategy prevents this case (see distribution of the number of stable binary values across all TWs).

When a camera moves, the number of its *active TWs* (its visible feature tracks), $\hat{T}_k$, decreases over time. We localise new binary features when their number is lower than a threshold χ with respect to the maximum number of features, F, localised and/or tracked, in a frame: $\hat{T}_k < \chi F$. To keep at most F active TWs, we limit the localisation outside a mask that eliminates the pixels in a small window centred at the current location $\mathbf{x}_{i,k}$ of a feature track. A new feature track is initialised for each new interest point that is successfully tracked in the next frame. We then update the representation of all active TWs with Eq. 3.7 and Eq. 3.8 only at frames with new binary features localised, instead of each frame, in order to avoid redundant computations.

4.3 Growing an adaptive tree

As the number of stable TWs increases over time, so does the computational cost for searching and matching stable TWs, for example via nearest neighbour search. To reduce these costs, we

[1]Note that feature tracks are not terminated in the current implementation.

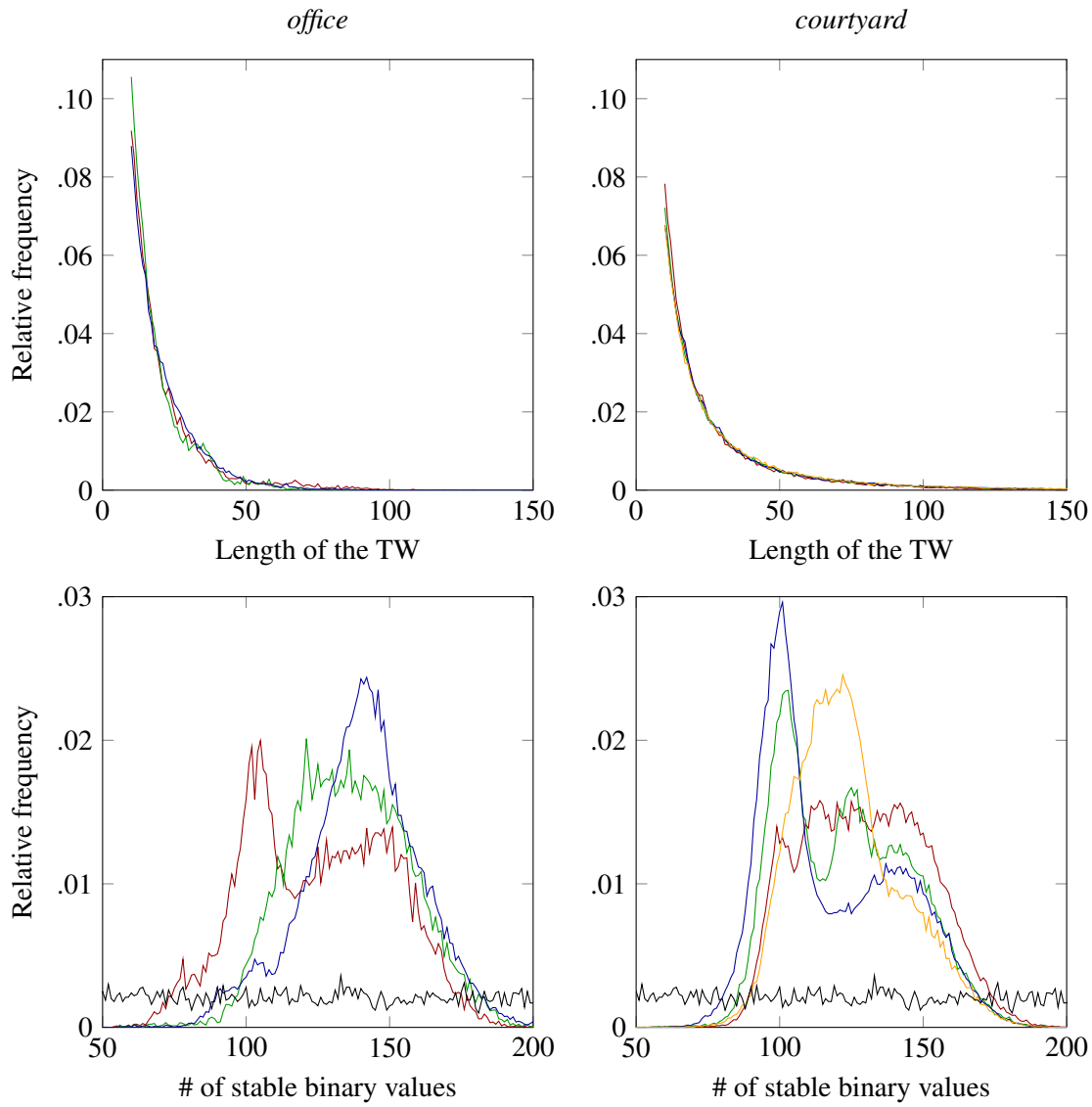


Figure 4.1: Distribution of the stable tracked words (TWs) based on their length (top) and distribution of the number of stable binary values across all TWs (bottom) for each sequence in the testing scenarios *office* and *courtyard*. The total number of stable TWs for *office* is 18376, 18247, 44829, for seq1, seq2, seq3, respectively; and for *courtyard* is 78574, 82470, 92401, 88500, for seq1, seq2, seq3, seq4, respectively. Note that there are stable TWs with length greater than 150 and up to 800 in *courtyard*, but we show only up to 150 for visualisation and comparison purposes. Note also that in the experiment we set the minimum length of the TWs, ρ, to be 10. We also include the distribution of the number of stable binary values across 10,000 TWs whose associated binary features are randomly generated from a Bernouilli distribution and lengths are randomly generated from a uniformly distribution in the range $[10, 100]$ (—). Legend: — seq1, — seq2, — seq3, — seq4.

organise stable TWs in a Adaptive search Tree of Stable TWs (ATST). However, a tree structure may find a set of feasible matches that is not as close as to the set of all feasible matches obtained without the tree structure (completeness) [79].

To achieve a trade-off between efficiency and completeness, we limit the depth of the tree by

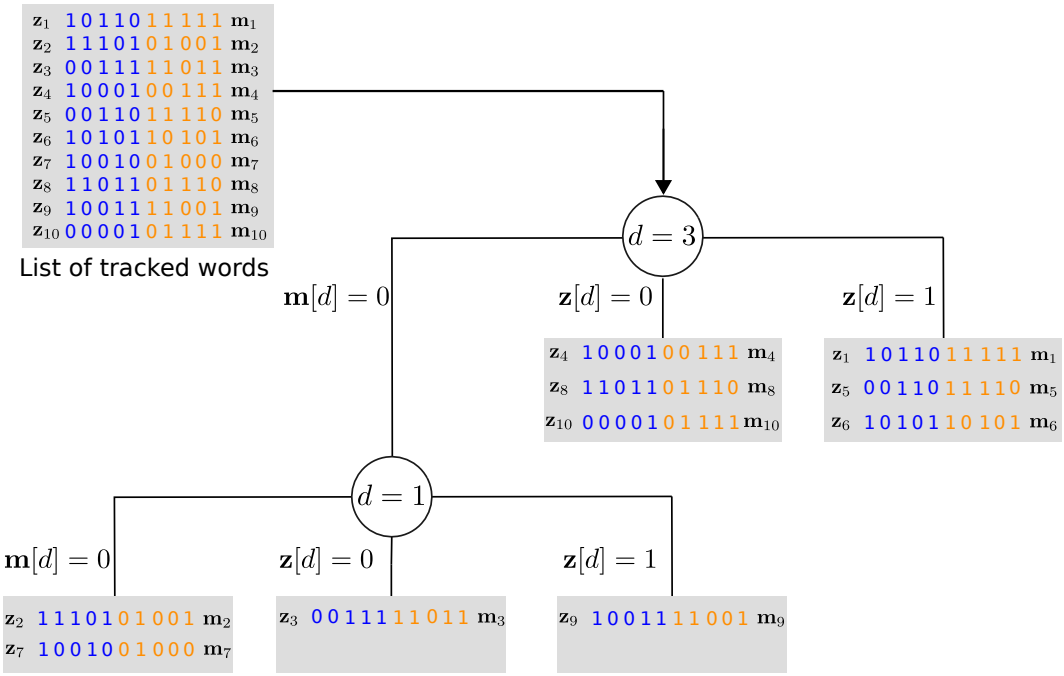


Figure 4.2: Ternary assignment of stable tracked words (TWs) with the optimal partitioning index (d) estimation for a balanced tree. Each parent node assigns a TW to one of the three child nodes (either a node with a condition, *i.e.* circle, or a leaf node, *i.e.* grey rectangle) based on the value at the estimated optimal partitioning index of the stability vector, $\mathbf{m}$ (in orange), and of the vector with the most frequent values over time, $\mathbf{z}$ (in blue). The procedure is recursively done until each TW is assigned and stored into a leaf node. In this example, a leaf node can store a maximum of three TWs, otherwise it is converted into a parent node and TWs are assigned to the new three child nodes (second level).

allowing a maximum number of TWs, N, to be stored in the leaf nodes, *i.e.* nodes that do not have any other node depending on them. When the number of stable TWs in a leaf node exceeds N, we convert the node into a parent node with three child nodes (new leaf nodes), where the parent node contains the splitting condition to assign stable TWs to any of the child nodes and the child nodes contain subsets of the stable TWs. Aiming to build a balance tree (*i.e.* stable TWs are evenly assigned to the child nodes), we adopt a bit selection strategy that accounts for the stability vector, $\mathbf{m}_i$, to estimate the optimal partitioning index [79]. We compute the optimal partitioning index, $d_j^\star$, at the j-th leaf node, which contains N_j stable TWs, as

$$d_j^\star = \arg\min_d \left| \frac{2}{3} - \sum_{j=1}^{N_j} \frac{\mathbf{m}_{j,d}}{N_j} \right| + \left| \frac{1}{3} - \sum_{j=1}^{N_j} \frac{\mathbf{m}_{j,d} \wedge \mathbf{z}_{j,d}}{N_j} \right|, \tag{4.1}$$

where $\wedge$ is the logical AND operator. The bit selection strategy considers the binary value at index d in both $\mathbf{z}_i$ and $\mathbf{m}_i$: if $\mathbf{m}_{i,d} = 0$, the i-th TW is assigned to the first child node, if $\mathbf{m}_{i,d} = 1 \wedge \mathbf{z}_{i,d} = 0$, the TW is assigned to the second child node, and if $\mathbf{m}_{i,d} = 1 \wedge \mathbf{z}_{i,d} = 1$, the TW is

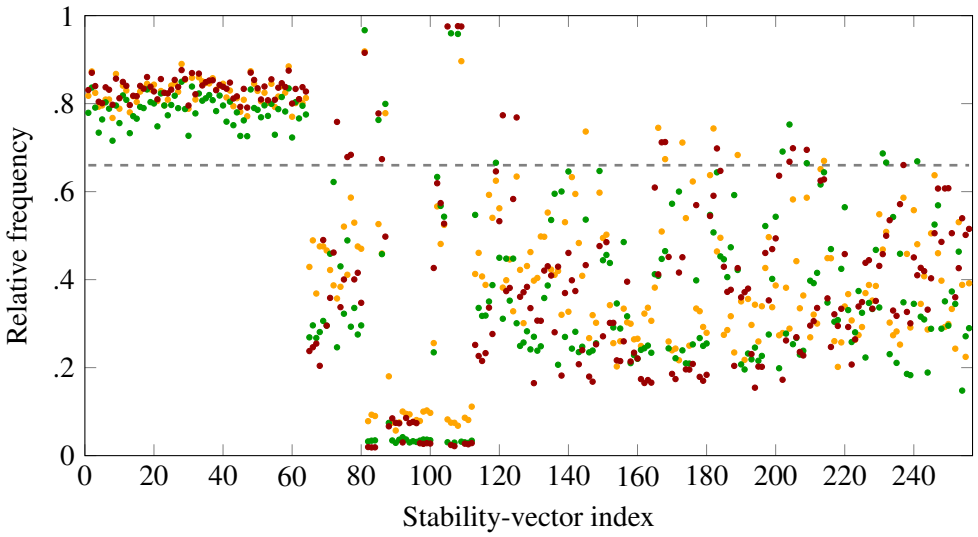


Figure 4.3: Identification of candidate stability-vector values as optimal partitioning index for achieving a balanced ternary tree (relative frequency close to 0.66, dashed grey line), in sampled sequences of three testing scenarios: ● *office* (seq1), ● *gate* (seq2), ● *courtyard* (seq4). The total number of TWs for each sequence are 18376, 11286, and 88500, respectively.

assigned to the third child node. Figure 4.2 shows an example of the ternary assignment for a set of stable TWs in a tree with a 2-level depth using the optimal partitioning index strategy. If $\mathbf{m}_i$ followed the Bernouilli distribution with probability 0.5, then about half of the TWs would be assigned to the first child node, while the other half would be almost equally split between the other two child nodes. Figure 4.3 shows that in practice the distribution of the stable values for all the TWs in sampled sequences from three different testing scenarios is not close to 0.5. While the first binary tests are the most stable, *i.e.* a normalised frequency to be 1 higher than 0.8, some tests around the element 100 are lower than 0.2 and most of the remaining binary values vary between 0.2. and 0.8. Note that the same pattern was observed in other testing sequences.

At every frame with a localisation of new binary features, the tree is adaptively updated by inserting new TWs, removing short TWs ($L_i < \rho$, where ρ is the minimum length of a feature track) and re-assigning TWs whose representations changed since the last frame with the localisation of new binary features.

For a query TW, $\mathbf{w}_q$, the camera efficiently searches for the leaf node j^* exploiting the optimal partitioning index for each level of the tree. To account for the stability information, we match $\mathbf{w}_q$, with each TW, $\mathbf{w}_i$, in the leaf j^* using the selective weighted Hamming distance (see Eq. 3.17). Figure 4.4 shows an example of searching and matching a query TW in a 2-level tree.

To limit erroneous matches, we first consider only those matches whose distances are smaller

than a dynamic threshold $\hat{\gamma}_{i,q}$ that accounts for the stability of the two TWs. We first define the normalised stability length of a TW i as $p_i = M_i/D$ (likewise p_q for a TW q to be matched), where D is the vector dimensionality, and then we compute

$$\hat{\gamma}_{i,q} = \min(p_i, p_q)\gamma. \tag{4.2}$$

The value of γ is usually defined by the separation of matching and non-matching point distributions when there is no stability information (see Section 4.3).

To avoid degenerative cases where $M_i \to 0$, *i.e.* the descriptor is bearing too little information losing its discriminative capability, we do not compute the distance between the two TWs if $\min(p_i, p_q) < 0.125$, *e.g.* $\min(M_i, M_q) < 32$ when $D = 256$[2].

To avoid ambiguities, we then discard matches whose distance ratio between the closest and the second closest TWs given a query TW is lower than δ (Lowe's ratio test [55, 61], usually $0.6 \leq \delta \leq 0.8$). Moreover, we discard matches whose either TW was already retained with a smaller distance to enforce only unique matches.

Unlike exhaustive (brute-force) or linear (nearest neighbour) search approaches, whose computational costs grow proportionally with the number of TWs, this hierarchical searching and matching strategy limits the domain of TWs that can be matched. Figure 4.5 shows how the number of TWs and the height of the tree grow over time, and compares the matching speed of using the tree to simply using an incremental list during place recognition at frames when ATST is updated. We compare the growth of the tree over time when using either a binary or a ternary search tree and with different maximum numbers of stable TWs stored in the leaf nodes, *e.g.* $N \in \{50, 100, 250\}$. Note that the tree rapidly increases in the first 100 frames and then remains stable for long periods before increasing again, while the growth is less frequent when increasing N. As expected, the matching time of the list of TWs increases proportionally with the number of TWs, for example reaching about 10 seconds when the number of TWs is about 12,000, whereas the matching time of our tree is in the order of 100ms on average and lower than 0.5 seconds most of the time.

However, when searching and matching a query TW within the tree, finding the correct match is highly affected by the partitioning index for each tree level, *i.e.* this may lead to a leaf node that does not contain the best match. Another match with a similar TWs may be found, resulting

[2]D-BRIEF [101] is the shortest binary descriptor, with $D = 32$.

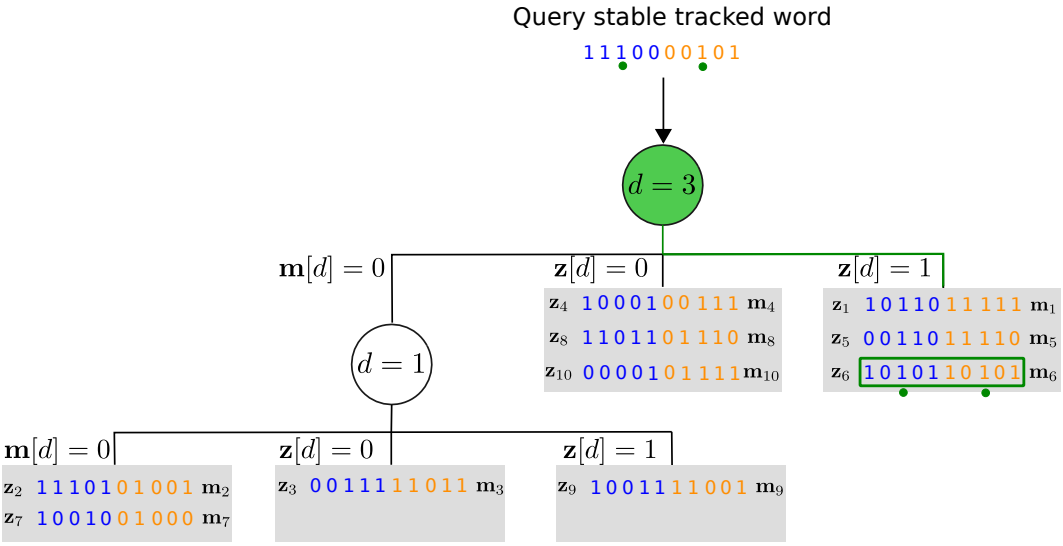


Figure 4.4: Searching and matching a query tracked word in a ternary search tree with a 2-level depth. The third positions (green dot) of the vector of the most stable binary values (orange) and of the vector of the most frequent binary values (blue) are used to search at which leaf node of the first tree level the query word should be redirected to find a match. In the retrieved leaf node after the search, the query word is matched against all the tracked words within the leaf node using the selective weighted Hamming distance (see Eq. 3.17). The best match is denoted with a green rectangle (the distance is 1 in this example).

in a geometrically incorrect match and leading to successive wrong estimations. XC-PR adopts a geometric verification step to further filter out possible erroneous matches, as discussed next.

4.4 View selection and place recognition

Let two cameras move independently in an unknown environment, while exchanging their TWs to identify previously seen places. Each camera operates independently and initialises its tree before sharing TWs. The initialisation lasts at least η frames, defined as

$$\eta = \max\left(3\rho, (\bar{K}/3 > 1)\bar{k}\right), \tag{4.3}$$

where $\bar{K}$ is the number of frames with new localised binary features, $\bar{k}$ is the index of the last of these frames, and ρ is the minimum length of a feature track.

After the initialisation, a camera shares a subset of TWs along with the interest points of corresponding binary features from a selected frame to query the previously seen places of the other camera. XC-PR uses an adaptive temporal window, $[t^\star, t]$, to select the frame $k^\star$ with the highest number of binary features from corresponding stable TWs that are no longer active in the current frame t. The length of the temporal window is a trade-off between the longest active TW

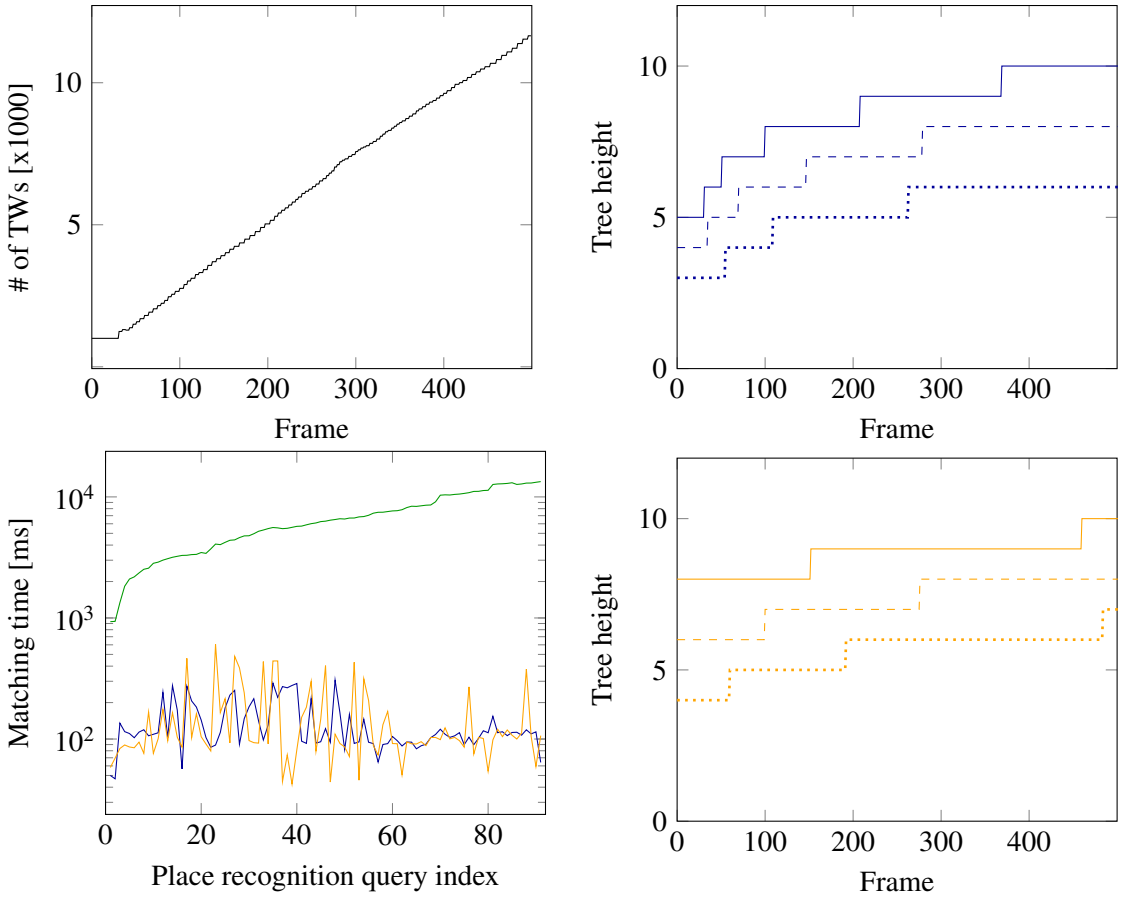


Figure 4.5: Analysis of the growing number of stable TWs (top-left), tree height (top-right, bottom-right), and comparison of the matching time (bottom-left) for the first 500 frames on *courtyard seq1* when using ATST as a binary search tree (—), ATST as a ternary search tree (—), and an incremental list (—). Note that the increasing number of TWs is the same for ATST and the incremental list (top-left). Also note how the height of the tree increases when varying the maximum number of TWs store in each leaf node: — 50, - - - 100, ···· 250.

at the current frame t and a maximum number of frames, Λ:

$$L^\star = \min\left(\arg\max_{L_i}\{L_i|\forall i, k_i = t\}, \Lambda\right), \tag{4.4}$$

with $t^\star = t - L^\star + 1$.

Therefore, let $\mathcal{W}_t = \{\mathbf{w}_i | t^\star < k_i < t\}$ be the subset of non-active stable TWs whose last frames k_i are within the adaptive temporal window, and $v_{i,k} \in \{0,1\}$ the visibility of the i-th stable TW in frame k. The camera selects the frame with highest number of visible TWs as

$$k^\star = \arg\max_k \sum_{i \in \mathcal{M}_t} v_{i,k}, \quad t^\star < k < t, \tag{4.5}$$

and then shares the subset of TWs along with the corresponding interest points,

$$\mathcal{Q}_t = \{(\mathbf{w}_i, \mathbf{x}_{i,k^\star}) | \nu_{i,k^\star} = 1\}. \tag{4.6}$$

A camera shares this information for frames when the number of feature tracks is substantially reduced (see Section 4.3).

To recognise a previously seen place, the other camera searches and matches the query TWs locally in its tree (see Section 4.3), and selects the candidate view with the highest number of binary features from corresponding matched TWs. To select this previous view, XC-PR uses Eq. 4.5 with the condition: $\forall k < t$, as there are no temporal relations between the cameras. We then geometrically validate the epipolar constraint between the interest points of matched TWs from the selected view and those of the received TWs from the other camera. We estimate the fundamental matrix, subject to a minimum of eight matched TWs [38], and we discard outliers through random sample consensus [31]. For each received subset $\mathcal{Q}_t$, the camera then acknowledges with the recognised place, $k^\star$. Figure 4.6 shows examples of matches between the interest points of query TWs from a selected view in one camera and interest points of TWs from the correctly recognised place.

4.5 Summary

In this chapter, we presented XC-PR, a novel Cross-Camera Place Recognition approach that identifies previously seen places across cameras, while the cameras move independently in an unknown environment. Each camera forms locally distinctive and compact descriptors, referred to as stable tracked words, that preserve the most persistent values from accumulated binary descriptors of local features as observed in multiple frames. At automatically selected frames, each camera independently exchanges selected descriptors to recognise previously seen places in another camera. To efficiently search and match query descriptors, each camera independently organises the descriptors in an Adaptive Tree of Stable Tracked Words that is selectively updated while the camera moves. XC-PR recognises a place by geometrically validating the previous frame with the highest number of binary features corresponding to matched descriptors.

While the proposed approach is designed for a pair of cameras, XC-PR might be applied to a scenario with more than two cameras, in a pairwise way. However, XC-PR may not easily scale to a high number of cameras (*e.g.* > 5), and substantial re-designs might be necessary, borrowing

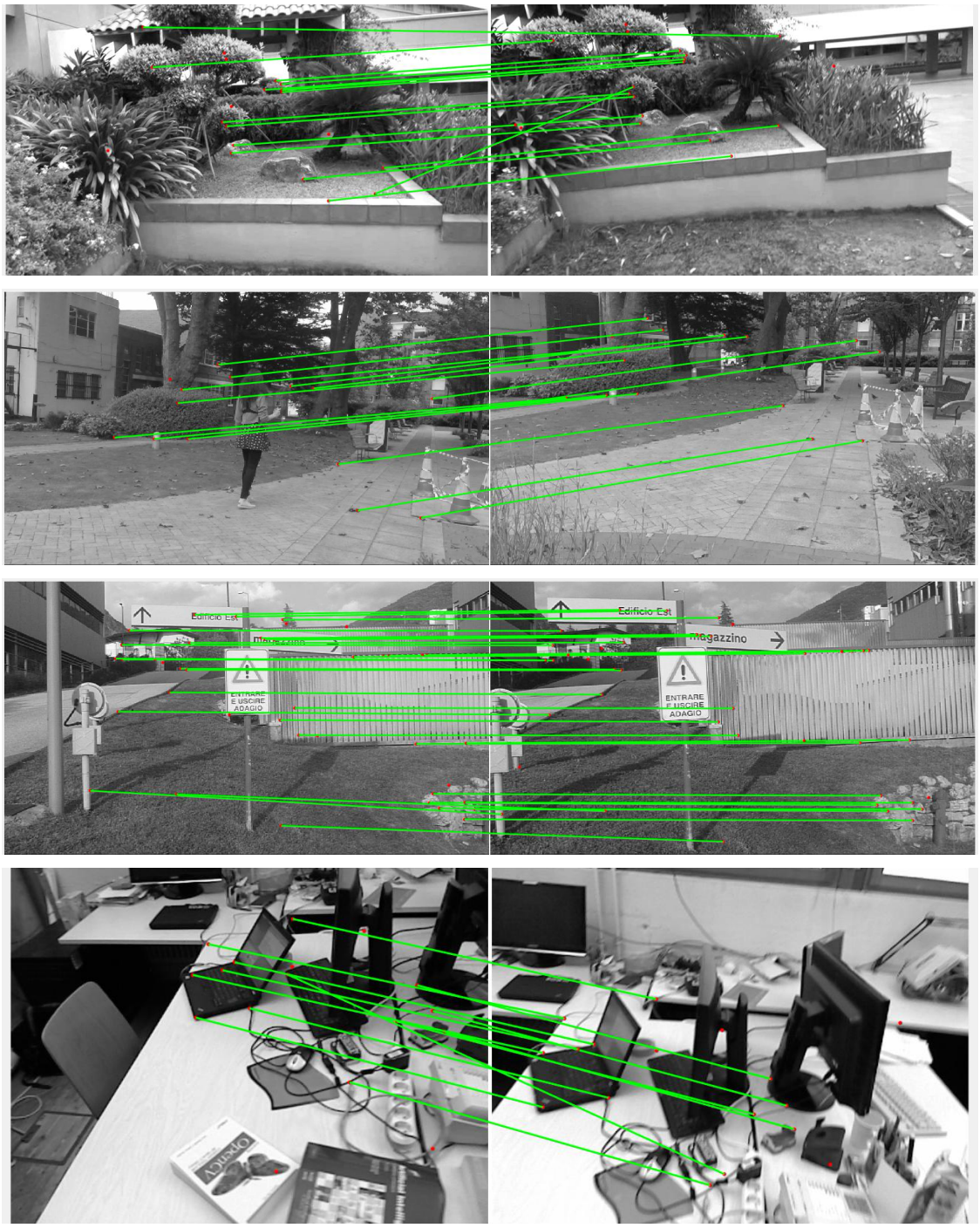


Figure 4.6: Sample of matches (— inliers) between the interest points (■) of corresponding query tracked words from a selected view in one cameras (left) and the interest points of corresponding tracked words found in a previous place of the other camera (right). From top to bottom: *courtyard 1|2*, *backyard 1|4*, *gate 1|4*, and *office 1|2*.

concepts from alternative distributed approaches [19].

Chapter 5

Experimental validation

5.1 Introduction

In this chapter, we evaluate and analyse the performance of the local image and spatio-temporal features introduced in Chapter 3, and the cross-camera place recognition (XC-PR) approach introduced in Chapter 4.

In Section 5.2, we evaluate the proposed multi-scale binary descriptor, MORB, against other local image binary features on standard image matching benchmarks. In Section 5.3, we propose a novel evaluation of the spatio-temporal features that recalls the evaluation of local image features but matching the features across short image sequences instead of single images. We show the performance of MST, the spatio-temporal multi-scale descriptor, against the spatio-temporal features and the bag of visual words approach. As these features are generic for different underline local image descriptors, we also compare the spatio-temporal features using different binary descriptors. In Section 5.4, we evaluate the place recognition accuracy and speed of XC-PR when using two variants of the proposed Adaptive Tree of Stable Tracked Words (ATST), namely binary search tree and ternary search tree, compared to using an incremental list of tracked words and a frame-based bag of binary words approach adapted to our framework.

5.2 Image matching

We compare MORB with ORB [77] and with LATCH [51] using interest points detected with MORB. In this case, we refer to ORB and LATCH as cORB and oLATCH, respectively. As

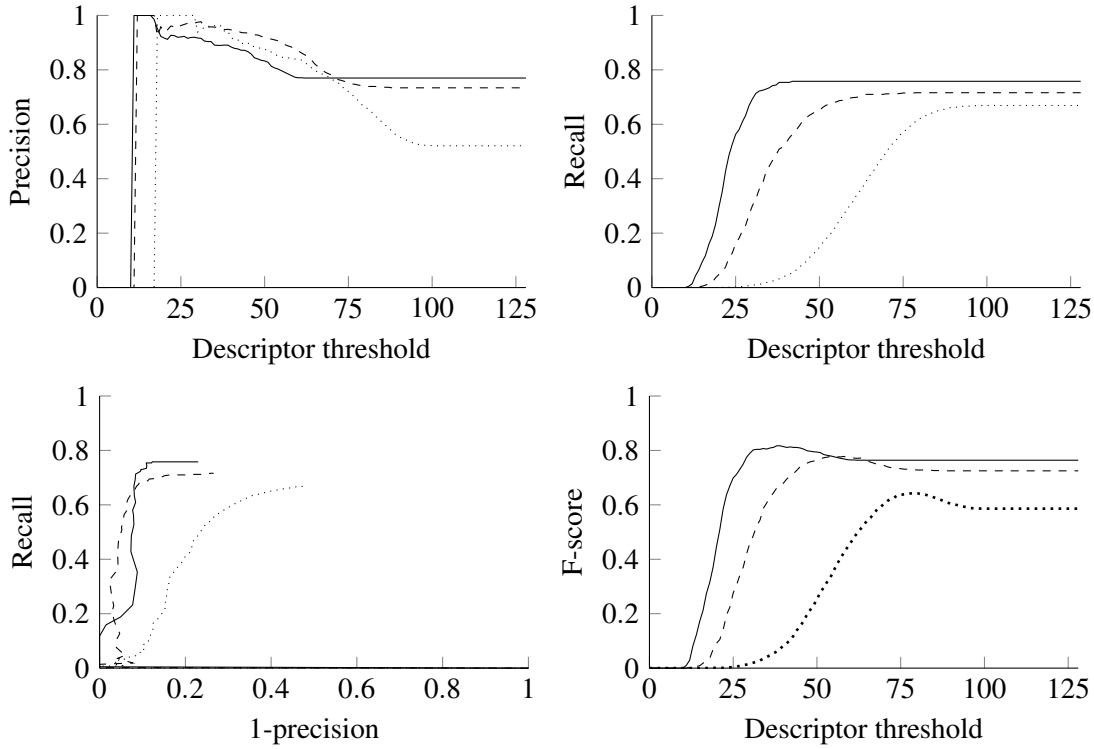
LATCH was paired with SIFT in [51], we also report the results of LATCH applied on interest points detected with SIFT (sLATCH). Furthermore, we report the results of ORB with its own detections and we test an all-to-all matching of independent ORB descriptors extracted for all scales (ORB-ALL).

In the detection phase, MORB uses the same settings (OpenCV 3.3 implementation) as ORB: the FAST threshold is 20, the patch size is $G = 31$, the number of scales is $S = 8$, the scale factor is $\lambda = 1.2$, and the kernel size $W = 7$ with $\sigma = 2$. We analysed the performance of MORB and ORB by varying F from 500 to 1500 with step 250, but we report only the results for $F = 1000$, similarly to LIFT [116], LDB [114], and SuperPoint [24], as we did not observe any significant performance changes[1].

As we propose a scale-aware nearest neighbour matching strategy for MORB, we evaluate ORB and LATCH with the nearest neighbour approach as *similarity matching* [61] on the Oxford Affine Covariance Regions Dataset (ACRD) [61], and on the *venice* set from Heinly's image matching [42] to consider only scale variations. We define a correspondence (as well as a correct match) as the pair of interest points with the lowest distance below 2.5 pixels after homography transformation (homographies are provided as ground-truth along with the dataset), with all interest points scaled up to the original scale, as suggested in Heinly's image matching [42]. To analyse the impact of the descriptor threshold, we vary γ from 0 to 128, (*i.e.* half of the size of the descriptor) and we then compute the number of matches $\mathcal{V}$ and the corresponding number of correct matches to generate precision and recall curves. To rank and compare methods, the area under the curve is also used recent evaluations [9, 26, 51].

Precision and recall can be analysed together through recall vs 1-precision curves [61] or the F_1-score. Here, we propose to evaluate the methods with the area under the F_1-score curve as we observed that computing the area under the recall vs. 1-precision curves with the nearest neighbour matching strategy can lead to a method ranking that is inconsistent with the ranking obtained with the more detailed area under the precision or recall curves (see Figure 5.1). In the recall vs 1-precision curve, a good method should not significantly decrease in precision and should keep a high recall, or keep a high recall even if the precision tends to zero. However, good methods in precision and recall may cover a smaller area than methods decreasing in precision and having a lower recall, thus resulting in lower performance. On the other hand, the F_1-score

[1] The default maximum number of features for each image is $F = 500$ in ORB implementation.



Performance measure	**Methods**		
	ORB	sLATCH	MORB
Area under *Precision* curve	**.76**	.63	.75
Area under *Recall* curve	.53	.35	**.63**
Area under *Recall vs 1-precision* curve	.15	**.19**	.13
Area under F_1-*score* curve	.58	.36	**.66**

Figure 5.1: Precision, recall, recall vs 1-precision, and F_1-score curves for the image pair *boat* 1 – 2. It can be noted in the table that using area under the recall vs 1-precision curve can lead to inconsistent ranking. Area under the F_1-score curve better preserves precision and recall behaviour as it is computed from their harmonic mean. Legend: - - - ORB, ···· sLATCH, — MORB.

can preserve the performance of precision and recall for evaluating the methods. We therefore refer to the area under the F_1-score curve as Nearest Neighbour Average F_1 score (NN-AF). While NN-AF evaluates the distinctiveness of the descriptor, we also compute the matching score (MS), *i.e.* the number of correct matches over the minimum number of features in common after homography transformation, with $\gamma = 128$, to measure the overall performance of the interest point detector and descriptor combined [24, 42, 57, 116].

Table 5.1 shows the NN-AF and MS results for each image pair in *venice* and in each set of the ACRD dataset. MORB outperforms the other descriptors in the three sets with either only scale variations (*venice*) or in-plane rotations and scale variations (*bark* and *boat*) as well as in other sets under other geometric and photometric transformations, except for illumination

changes (*leuven*). In the illumination case, sLATCH is the best performing method. As oLATCH performs similarly to cORB in *leuven*, the good performance of sLATCH are related to the interest points detected by SIFT [55] that is more invariant to illumination changes. Nevertheless, LATCH is sensitive to scale changes both with the ORB and the SIFT detector. Most of the NN–AF performance are supported by a similar or higher MS, showing the capability of MORB to find more correct matches than the other descriptors. We can also observe that cORB performs worse than ORB due to the discarded interest points that could be relevant for the matching. We proved the effectiveness of our cross-scale matching over ORB-ALL showing that the independence assumption of single descriptors across scales for each feature decreases the matching performance.

Figure 5.2 shows the area under the precision curves and the area under the recall curves in relation to the scale ratio between the image pairs in *venice*, *boat* and *bark*. While all ORB variants and MORB have similar precision performance, MORB outperforms in recall, thus estimating more correct matches than the other descriptors. As mentioned earlier, sLATCH and oLATCH perform poorly except when the scale change is small (scale ratio close to 1) where their performance is closer to that of ORB.

Table 5.2 shows the average running time (and standard deviation) for detection, description and matching of MORB features across all the testing image pairs. While the running time for detection and description at multiple scales is around 80 ms on average, the main bottleneck is the running time for matching MORB features between two images (about 2.5 secs on average), as the *set2set min dist* is quadratic with the number of scales, S^2.

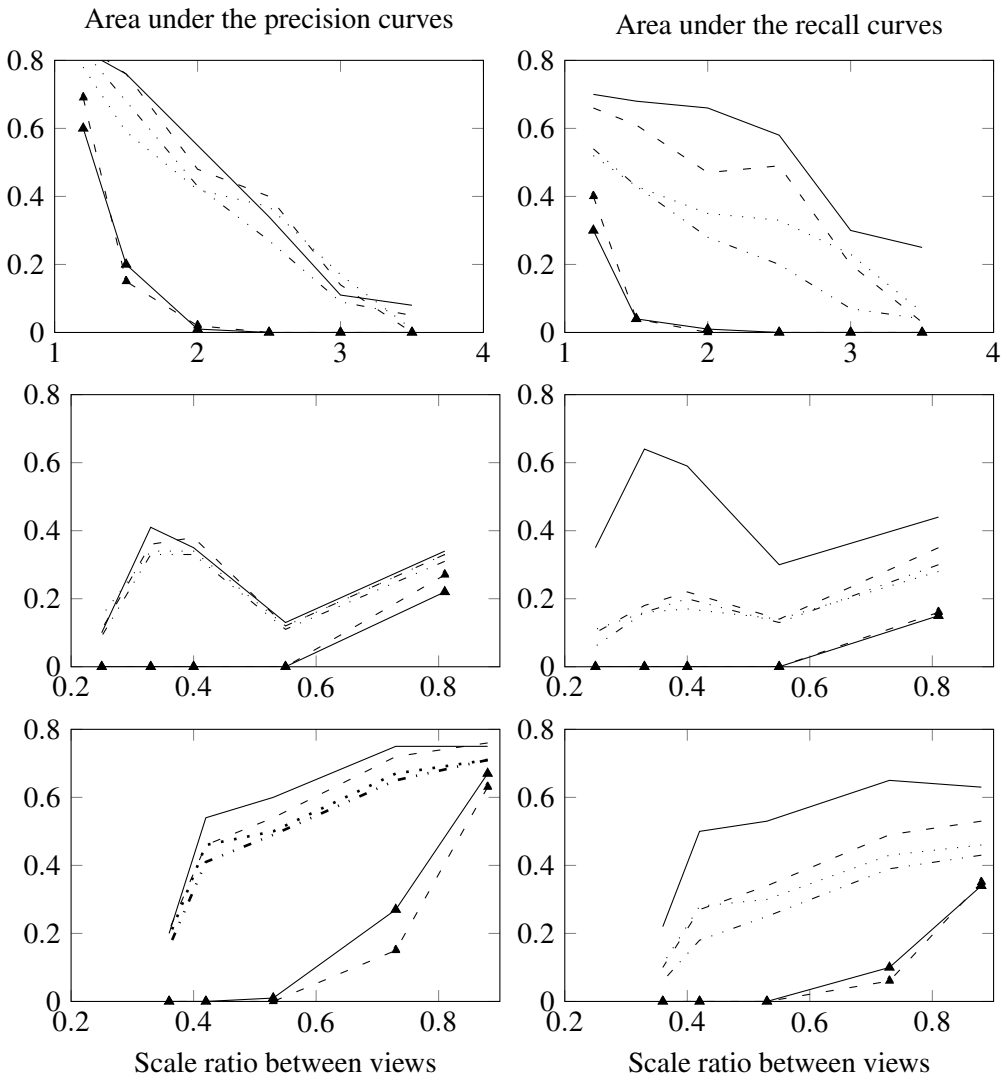


Figure 5.2: Area under the precision curves (left) and area under the recall curves (right) when increasing the scale ratio between image pairs in *venice* (top), *bark* (middle) and *boat* (bottom). While *venice* shows an increasing zoom in, *bark* and *boat* shows an increasing zoom out of the target images with respect to the reference image. Legend: - - ORB, - ·▲ sLATCH, · · · cORB, — ▲ oLATCH, - · · ORB-ALL, — MORB.

Table 5.1: Nearest Neighbour Average F_1 score (NN-AF) and Matching score (MS) for each image pair for each set of images. Best results in bold.

		NN–AF						MS					
		ORB	sLATCH	cORB	oLATCH	ORB-ALL	MORB	ORB	sLATCH	cORB	oLATCH	ORB-ALL	MORB
venice	1 – 2	.70	.42	.57	.35	.62	**.73**	**.57**	.45	.46	.41	.43	.51
	1 – 3	.61	.04	.45	.05	.49	**.69**	**.45**	.05	.33	.06	.30	.44
	1 – 4	.41	.00	.31	.00	.28	**.56**	.20	.00	.16	.00	.12	**.25**
	1 – 5	.35	.00	.25	.00	.17	**.38**	.12	.00	.09	.00	.05	**.14**
	1 – 6	.12	.00	.14	.00	.05	**.14**	.03	.00	.04	.00	.01	**.05**
	1 – 7	.01	.00	.02	.00	.02	**.09**	.00	.00	.01	.00	.00	**.03**
bark	1 – 2	.28	.14	.22	.12	.25	**.33**	.11	**.12**	.09	.08	.09	**.12**
	1 – 3	.09	.00	.09	.00	.08	**.15**	.03	.00	.02	.00	.02	**.04**
	1 – 4	.18	.00	.13	.00	.16	**.37**	.04	.00	.03	.00	.04	**.10**
	1 – 5	.13	.00	.11	.00	.12	**.37**	.03	.00	.03	.00	.03	**.10**
	1 – 6	.04	.00	.04	.00	.02	**.09**	.01	.00	.01	.00	.00	**.02**
boat	1 – 2	.58	.36	.50	.40	.50	**.66**	.46	.31	.41	.41	.36	**.48**
	1 – 3	.53	.06	.46	.11	.44	**.66**	.39	.06	.34	.14	.28	**.43**
	1 – 4	.36	.00	.32	.00	.28	**.51**	.22	.00	.20	.00	.15	**.30**
	1 – 5	.27	.00	.28	.00	.19	**.46**	.15	.00	.15	.00	.09	**.24**
	1 – 6	.08	.00	.09	.00	.05	**.16**	.04	.00	.04	.00	.02	**.07**
graffiti	1 – 2	.55	.43	.49	.33	.48	**.64**	**.46**	.39	.39	.33	.34	.45
	1 – 3	.27	.09	.21	.12	.21	**.33**	.20	.09	.16	.12	.14	**.23**
	1 – 4	.11	.02	.10	.03	.08	**.12**	**.08**	.02	.07	.04	.05	**.08**
	1 – 5	**.02**	.00	**.02**	.00	.01	.01	**.02**	.00	**.02**	.00	.01	.01
	1 – 6	.00	.00	**.01**	**.01**	.00	.00	.00	.00	**.01**	**.01**	.00	.00
wall	1 – 2	.48	.50	.44	.34	.46	**.65**	.36	**.50**	.33	.31	.32	.45
	1 – 3	.44	.35	.38	.26	.39	**.61**	.34	.34	.30	.26	.29	**.44**
	1 – 4	.23	.16	.22	.13	.21	**.36**	.14	.14	.14	.11	.12	**.21**
	1 – 5	.08	.04	.08	.04	.07	**.13**	.04	.04	.04	.03	.03	**.07**
	1 – 6	**.01**	.00	.00	**.01**	**.01**	**.01**	.00	.00	.00	.00	.00	**.01**
bikes	1 – 2	.71	.66	.62	.62	.66	**.76**	**.61**	.50	.51	.53	.49	.55
	1 – 3	.65	.63	.56	.58	.61	**.73**	**.54**	.52	.45	.47	.44	.50
	1 – 4	.53	.55	.44	.45	.55	**.67**	.38	**.46**	.31	.35	.35	.41
	1 – 5	.43	.51	.32	.34	.46	**.57**	.28	**.44**	.22	.26	.28	.34
	1 – 6	.35	.41	.25	.26	.37	**.48**	.20	**.36**	.15	.18	.20	.25
trees	1 – 2	.49	.30	.39	.36	.47	**.59**	.32	.21	.25	.26	.28	**.34**
	1 – 3	.41	.24	.33	.30	.40	**.55**	.22	.18	.18	.18	.21	**.27**
	1 – 4	.27	.15	.21	.23	.25	**.35**	.12	.10	.10	.13	.11	**.15**
	1 – 5	.21	.12	.16	.16	.23	**.30**	.09	.09	.07	.09	.09	**.11**
	1 – 6	.13	.07	.11	.13	.15	**.22**	.04	.06	.04	.06	.05	**.08**
leuven	1 – 2	.67	**.77**	.61	.59	.57	.70	.48	**.59**	.43	.42	.37	.44
	1 – 3	.60	**.73**	.54	.54	.52	.63	.38	**.55**	.36	.36	.31	.37
	1 – 4	.55	**.68**	.47	.51	.48	.62	.33	**.52**	.29	.31	.26	.34
	1 – 5	.51	**.64**	.41	.47	.45	.57	.28	**.48**	.24	.26	.23	.30
	1 – 6	.46	**.58**	.41	.44	.42	.53	.26	.43	.24	.26	.22	**.28**
ubc	1 – 2	**.93**	.76	.90	.88	.91	.89	**.90**	.57	.86	.86	.84	.77
	1 – 3	**.90**	.64	.86	.83	.87	.89	**.84**	.48	.79	.79	.77	.74
	1 – 4	.84	.50	.77	.74	.82	**.87**	**.78**	.35	.71	.72	.71	.71
	1 – 5	.70	.35	.63	.58	.69	**.79**	**.63**	.20	.57	.58	.58	**.63**
	1 – 6	.57	.26	.50	.45	.56	**.66**	**.51**	.17	.45	.46	.44	.48
ACRD avg.		.39	.29	.34	.28	.36	**.47**	.28	.23	.25	.23	.24	**.30**
Total avg.		.39	.27	.34	.26	.35	**.47**	.28	.21	.24	.21	.23	**.29**

Table 5.2: Average running times (seconds) for extracting and matching approximately 1000 keypoint per image for both ORB [77] and MORB over 100 runs on *boat* 1|2 (standard deviation in bracket). Note that detection for MORB includes removal of points too close at the borders at the lowest resolution of the image pyramid, the re-sampling at all scales, and the removal of duplicates. The orientation assignment is computed during the detection for both ORB and MORB.

Method	**Detection**	**Description**	**Matching**
ORB	0.019 (0.004)	0.011 (0.003)	0.068 (0.002)
MORB	0.050 (0.093)	0.029 (0.004)	2.530 (0.106)

5.3 Matching spatio-temporal features

We analyse and evaluate the performance of the spatio-temporal descriptors introduced in Chapter 3, namely T-D, T-DS, and MST. We first introduce a way to evaluate the matching of spatio-temporal features that extends the standard evaluation of local image features [61].

We compare our proposed multi-scale spatio-temporal descriptor, MST, against (i) SetDesc, the set of image-based binary descriptors of a feature track without reduction; (ii) T-D, extracted at a single scale with a reduction of the set of binary descriptors with only the temporally dominant bit approach (Section 3.3); (iii) T-DS, which complements T-D with a vector that contains the temporally stable bits (Section 3.3), (iv) LMED, which selects the single binary descriptor from SetDesc that has the least median distance compared to all other descriptors within the feature track [68]; and also (v) MST-S, which corresponds to our spatio-temporal descriptor without the stability vectors. Even if LMED was proposed for tracking binary features with a single camera, we analyse here its performance for cross-view matching.

To fairly compare all the descriptors, we obtain feature tracks with our approach and we then compute the corresponding descriptors. We set the parameters using values from related works or corresponding implementations: the FAST threshold is 25 [76], the block size for the grid is $w = 30$ [68]. To extract the multi-scale descriptor, we consider a pyramid of $S = 5$ scales [77] with a scale factor $\lambda = 1.15$. The patch size depends on the chosen image-based binary descriptor (*e.g.* $G = 31$ for ORB [77]). Features are tracked with the pyramidal Kanade-Lucas-Tomasi tracker [15] available in OpenCV using a window size of 21 pixels, 5 scales and maximum 30 iterations. We discard tracked features whose distance from the image boundaries at the coarsest level is less than half of G, which ensures the extraction of the descriptors at multiple scales. To reduce uncertainty in the triangulation, we enforce the feature track to be at least 5 frames long assuming that there is enough camera motion (translation) [38]. In addition to this, we set the radius of the non-maxima suppression for the grid-based detection to 3 pixels; and we detect new interest points every $n = 5$ frames[2] using a 7×7 masking window around the location of each existing feature track, for consistency with the radius of the grid.

Then, we compare MST against the method based on ORB-SLAM [68] for the processing of each sequence (feature track extraction and descriptor reduction) and a matching with the Bag of Binary Words (*e.g.* DBoW2 [33]). As last experiment, we show that the discussed spatio-

[2]We aim to reproduce the automatic keyframe selection strategy of Visual SLAM/Odometry methods (around 5-10 keyframes per second) [68, 28]

temporal features are generic and we compare them with a range of binary descriptors.

For all experiments, we use the most suitable dissimilarity measure for each descriptor when matching features. For SetDesc, we compute the minimum Hamming distance between all possible pairs of single descriptors between the sets of descriptors belonging to two different sequences (*set2set min dist* [40]). However, as finding the minimum across both scales and time is computationally expensive, we extract and match the sets only at the original scale. Note that we expect SetDesc extracted also at multiple scales to achieve higher matching performance than without scale. For T-D and LMED, we use the standard Hamming distance as dissimilarity measure, while we use the weighted Hamming distance (Eq. 3.17) for T-DS and MST as their descriptors contain the additional stability vector. Finally, we consider the cross-scale matching approach between single-scale descriptor pairs for MST-S and MST.

5.3.1 Evaluating spatio-temporal features

In Chapter 2, we reviewed the evaluation of local image features for the image matching problem. Precision and recall measures are generally used for features between image pairs with known ground-truth homographies [9, 42, 61]. Inspired by [61], we propose to extend the same evaluation to the matching of spatio-temporal features between short video streams.

As performance measures, we thus quantify the number of correct matches over the number of estimated matches (precision, P); the number of correct matches over the number of annotated reference correspondences (recall, R); and their harmonic mean (F_1-score). In addition, we can also determine the matching score (MS) and the average matching time per descriptor pair to evaluate the different spatio-temporal approaches. Therefore to compute P and R for feature point tracks, we consider two approaches to annotate reference correspondences: one exploits depth images when available with a given dataset, and the other uses multi-view geometry [38]. Both approaches assume the camera poses and calibration data are available with the dataset, *e.g.* provided by a motion capture system or by a Structure-from-Motion pipeline (*e.g.* COLMAP [83]).

When depth images are available for a pair of sequences acquired with an RGB-D camera, we relate each RGB pixel to its corresponding depth pixel. Using projective geometry [38], we reconstruct the 3D structure of the scene in a common reference system. We can then determine spatio-temporal features for each video stream as well as reference correspondences[3]. For each

[3]If the RGB and depth streams are acquired with different sampling rates, we consider the same depth image for two RGB images that are temporarily the closest to the depth image.

Figure 5.3: A sample frame for each sequence of the four sets. Note the differences in viewpoint and/or scale between sequences within the same set.

spatio-temporal feature, we compute a 3D location as the median of the set of 3D points estimated from the back-projection of the image locations and properly scale them using the associated values in the depth images. The median helps to remove false 3D estimations caused by noise or errors in the tracking of the spatio-temporal features. After obtaining a set of reconstructed 3D points for each video stream, we apply a brute force approach between the two sets and we then define the reference correspondences as the set of all 3D point pairs whose Euclidean distance is lower than 3 cm.

When depth images are not available, we reconstruct the 3D point associated to each feature track of one image sequence using multi-view geometry [38] given the absolute camera poses. We then geometrically verify that the projection of the point into the second view is within the image borders for at least five frames. Then, we compute the root mean square residual between feature track pairs from the two views and validate only pairs whose root mean square residual is smaller than 5 pixels. We determine the number of unique correspondences (*i.e.* one feature track cannot be paired with more than one in another view) using the nearest neighbour approach.

To evaluate spatio-temporal features, we use pairs of sequences, captured with hand-held cameras, from publicly available datasets: TUM-RGB-D SLAM [94]; *courtyard*[4] [121]; and *gate*, a dataset we collected and make available to the research community. Figure 5.3 shows sample frames for each sequence pair. TUM-RGB-D provides calibration data, camera poses obtained with a motion capture system, and RGB and depth streams. *courtyard* and *gate*, instead, contain only RGB streams and calibration data, therefore we obtain their cameras poses with COLMAP [83].

From TUM-RGB-D SLAM we use two clips of 50 frames (640×480 pixels) with sufficient

[4] `drone.sjtu.edu.cn/dpzou/project/coslam.php`, accessed: March 2018

overlap from *desk* (with similar motion) and *office* (cameras move in opposite directions)[5]. We select these two sequences because they contain enough texture for detecting and tracking features, and loop closures or different camera motions around the same scene. The two clips are selected in such a way that they simulate the motion of two cameras looking at the same portion of the scene from different viewpoints. From the first and fourth video of *courtyard*, we select the first 50 frames (800×450 pixels) after sub-sampling the videos from 50 to 25 fps. We select the first 100 frames (1280×720 pixels) of the four sequences of *gate* after down-sampling the video to 10 fps from 30 fps. We pair the first sequence with each of the other three sequences and we refer to each pair as *gate-1*, *gate-2*, *gate-3*, respectively.

5.3.2 Multi-scale temporal feature versus spatio-temporal features

In this experiment, we analyse MST and compare it against the spatio-temporal features SetDesc and LMED, and the previously proposed T-D and T-DS.

Figure 5.4 shows the percentage of survived trajectories, feature tracks that are discarded because of the short length, and feature tracks discarded by geometric tests as processed by MST for each sequence pair. The total number of feature tracks is ∼135,000. The high number of feature tracks denotes the frequent re-localisation of many interest points. Because of the short length, the method discards more than 50% of the feature tracks in most of the sequences except *desk* where the camera moves slowly. In *gate-3*, the geometric tests invalidate ∼25% of the feature tracks due to camera shaking.

Figure 5.5 compares the matching performance of MST with MST-S and the spatio-temporal features when varying F to quantify the impact of the number of feature points localised in the first frame or during the re-detection. For *courtyard*, *office*, and *gate-3*, MST outperforms other descriptors independently of the number of localised features. In *desk*, where the scene has low texture and small geometric variations, SetDesc achieves the best performance. When $F = 500$, the performance of MST-S and MST is close to SetDesc, while when increasing F the performance of MST converges to that of T-DS showing that the multi-scale is not important in this scenario. We can also observe that, unlike the behaviour of the other approaches, the performance of SetDesc increases when $F = 1000$ in *gate-2* and *gate-1*. In *gate-2* SetDesc achieves the highest F_1-score. Overall, we can observe that increasing the maximum number

[5]From *office*, we select the frames from 114 to 163 and from 2,305 to 2,354. From *desk*, we select the frames from 97 to 147 and from 390 to 340.

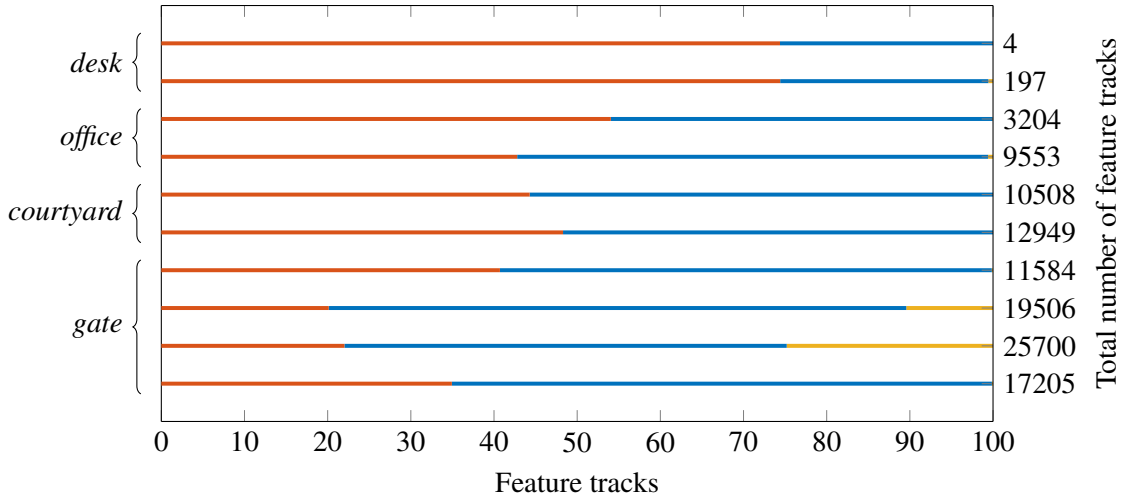


Figure 5.4: Percentage of surviving feature tracks (■) after discarding tracks whose length is shorter than five frames (■) or geometric tests (■). Note that the number of feature tracks invalidated by the geometric tests are less than 0.1% for most of the sequences.

of features localised or re-detected does not result in an increase of the performance, but on the contrary the performance tends to decrease in most of the sequence pairs for most of the approaches. For fair comparison, we do not fine-tune the number of features and we set $F = 2000$ across all sequences for the last comparison results.

Table 5.3 compares the matching performance of spatio-temporal descriptors extracted from the feature tracks. The number of reference correspondences is 1280 for *desk*, 2623 for *office*, 3448 for *courtyard*, 2427 for *gate-1*, 1357 for *gate-2*, 2378 for *gate-3*. We can observe that the additional stability vector of T-DS and MST leads to higher recall but lower precision than T-D and MST-S. Moreover, the multi-scale representation, MST-S and MST, allows to improve the performance of the proposed temporal reduction, *i.e.* T-D and T-DS. MST outperforms other approaches in terms of recall across all sequence pairs except *desk* that contains sequences with limited motion in the same direction and similar viewpoint in an indoor environment with low texture. The higher recall also influences the performance of the F_1-score except for *gate-2* where the stability vector allows to estimate almost twice the number of matches with several false positives (85/584 for MST vs 60/319 for MST-S), considerably decreasing the precision. We can observe that due to the severe change in viewpoint between the cameras, *office* and *gate-2* are the most challenging sequence pairs with recall lower than 12% for all approaches.

We now evaluate the spatio-temporal features without the geometric tests, but still filtering out short feature tracks (see Figure 5.4). Table 5.4 shows, for each spatio-temporal feature and for each sequence pair, the difference between the F_1-score with and without geometric tests. We

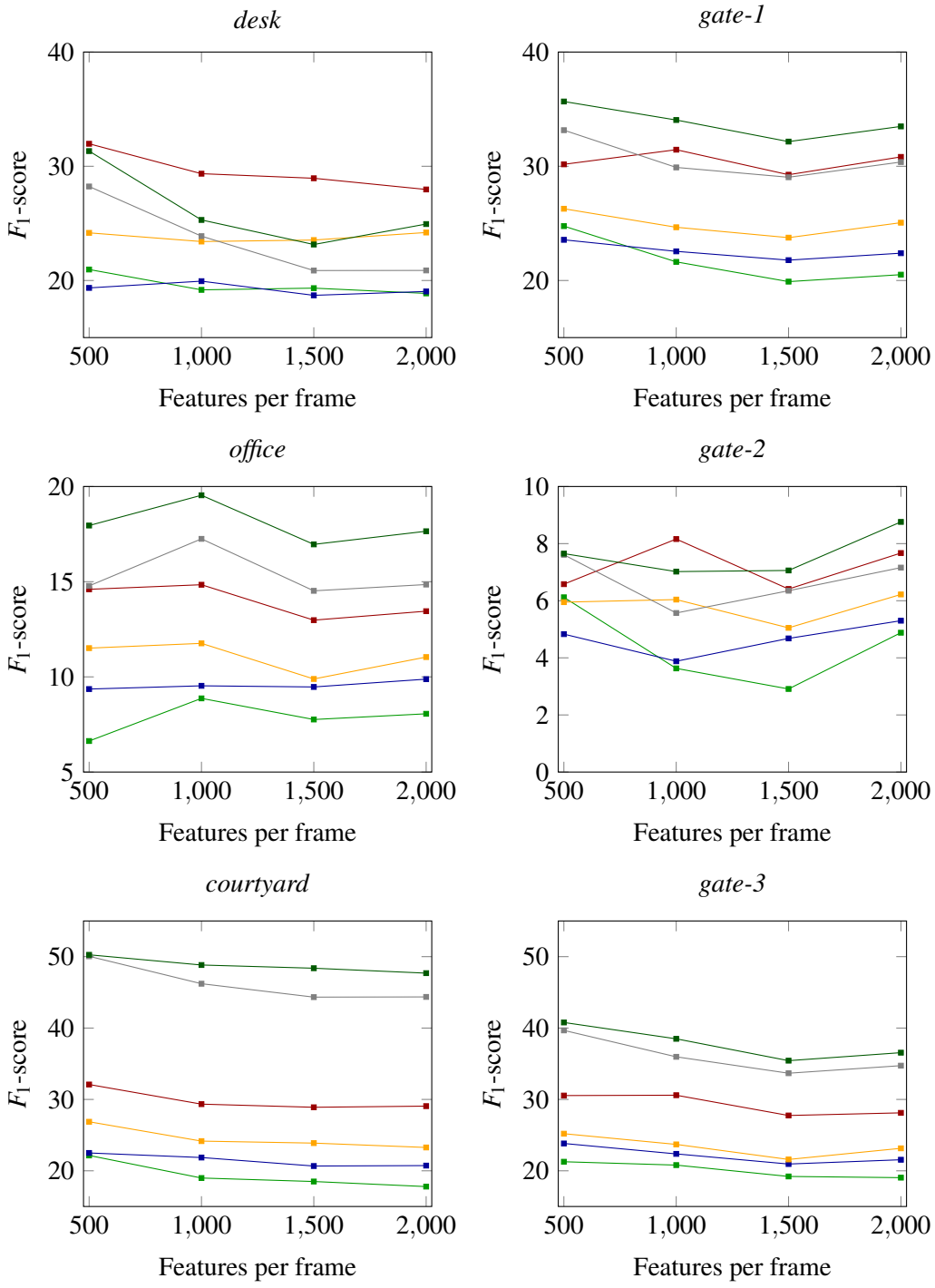


Figure 5.5: Accuracy (F_1-score) by varying the maximum number of features per frame using ORB [77]. The number of features per frame depends on the localisation in the first frame and re-detection. Note the different y-axis scales. Legend: SetDesc, LMED, T-D, T-DS, MST-S, MST.

can observe that adopting the geometric tests has a minimal impact on the accuracy for SetDesc, LMED, T-D, and MST across all sequence pairs, while T-DS and MST-S are the most sensitive to this step as their accuracy decreases up to 6% and less than 3% in F_1-score, respectively.

Table 5.3: Matching results with the nearest neighbour strategy and Lowe's ratio test using ORB features. Best results in bold, second best in italic.

Sequence	**Method**	**Performance measures**			
		Number of matches	Precision	Recall	F_1-score
desk	SetDesc	444	**54.28**	**18.84**	**27.97**
	LMED	321	*47.04*	11.81	18.87
	T-D	328	46.65	11.96	19.04
	T-DS	481	44.28	16.65	24.20
	MST-S	388	44.85	13.60	20.88
	MST	533	42.40	*17.67*	*24.94*
office	SetDesc	560	*38.21*	8.16	13.45
	LMED	453	27.37	4.73	8.06
	T-D	454	33.48	5.79	9.88
	T-DS	692	26.45	6.98	11.04
	MST-S	541	**43.44**	*8.96*	*14.85*
	MST	834	36.57	**11.63**	**17.65**
courtyard	SetDesc	853	73.27	18.13	29.06
	LMED	632	57.44	10.53	17.79
	T-D	671	63.64	12.38	20.73
	T-DS	1021	50.93	15.08	23.27
	MST-S	1214	**85.17**	*29.99*	*44.36*
	MST	1610	*74.91*	**34.98**	**47.69**
gate-1	SetDesc	895	**57.21**	*21.10*	*30.82*
	LMED	693	46.18	13.19	20.51
	T-D	700	50.00	14.42	22.39
	T-DS	1036	41.89	17.88	25.06
	MST-S	892	56.50	20.77	30.37
	MST	1293	48.18	**25.67**	**33.49**
gate-2	SetDesc	338	**19.23**	*4.79*	*7.67*
	LMED	282	14.18	2.95	4.88
	T-D	265	16.23	3.17	5.30
	T-DS	508	11.42	4.27	6.22
	MST-S	319	18.81	4.42	7.16
	MST	584	14.55	**6.26**	**8.76**
gate-3	SetDesc	880	*52.05*	19.26	28.12
	LMED	668	43.41	12.20	19.04
	T-D	741	45.34	14.13	21.55
	T-DS	1112	36.33	16.99	23.15
	MST-S	1095	**55.07**	*25.36*	*34.73*
	MST	1562	46.09	**30.28**	**36.55**

Figure 5.6 shows correct matches obtained with MST. Reconstructed 3D points are re-projected in pairs of selected frames for *gate-1*, *gate-2*, and *gate-3* with changes in both scale

Table 5.4: Difference between the F_1-score (%) of spatio-temporal features when extracting feature trajectories with and without 3D geometric tests.

Method	**Sequence pair**					
	desk	*office*	*courtyard*	*gate-1*	*gate-2*	*gate-3*
SetDesc	.00	-.12	-.01	.00	-.03	.10
LMED	-.02	-.01	-.08	-.01	-.01	-.16
T-D	.00	.00	.00	.00	.00	.00
T-DS	-4.80	-3.00	-3.10	-3.90	-5.20	-6.60
MST-S	-1.70	-.89	-.75	-1.40	-1.50	-2.60
MST	-.21	-.01	-.15	-.16	-.09	-.26

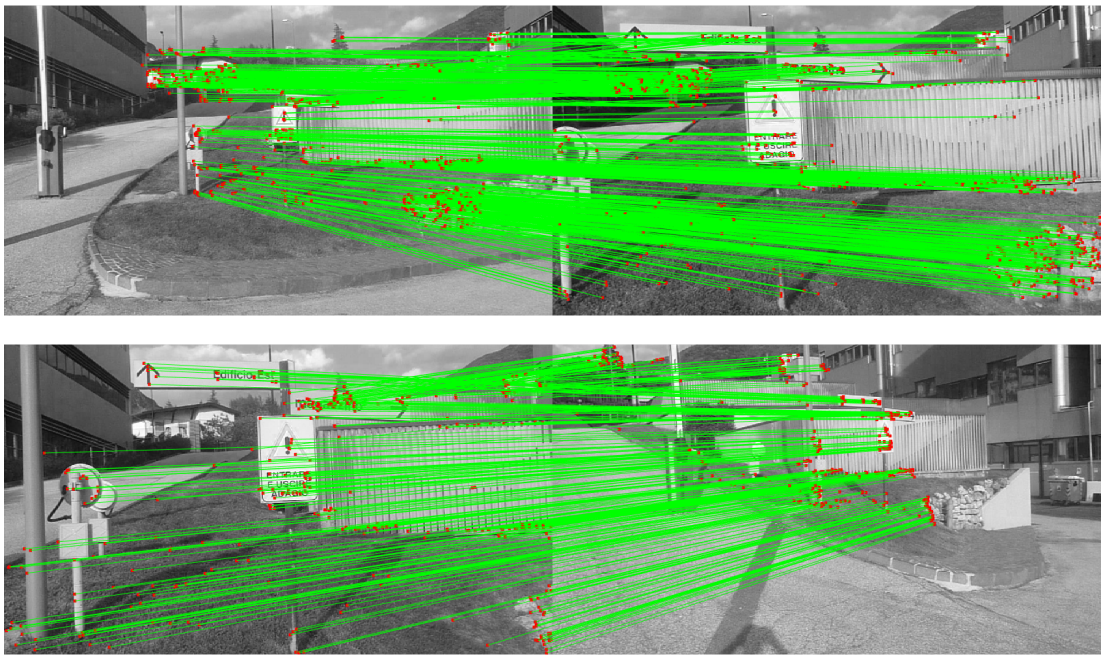


Figure 5.6: Correct matches (green lines) with MST by re-projecting the 3D points (red dots) in a selected pairs of frames. Top: scale difference in *gate-3*; bottom: viewpoint and scale difference in *gate-1*.

and viewpoint. Figure 5.7 quantifies the maximum viewpoint angle for MST features when estimated within each sequence and when correctly matched across cameras, for all sequence pairs. The viewpoint angle is computed using the cosine formula between the reconstructed 3D point and two camera locations, where the 3D point is observed. For each MST feature, we estimate the angle between each pair of views where the corresponding 3D point is visible, and we then find the view pair with the maximum angle. Most trajectories can handle up to 10 degrees of viewpoint difference, while there are features that can handle differences of up to 30 degrees. As feature tracking is performed with a validation strategy based on the image-based binary descriptor, the maximum viewpoint variation is constrained by the limitation in the geometric variations

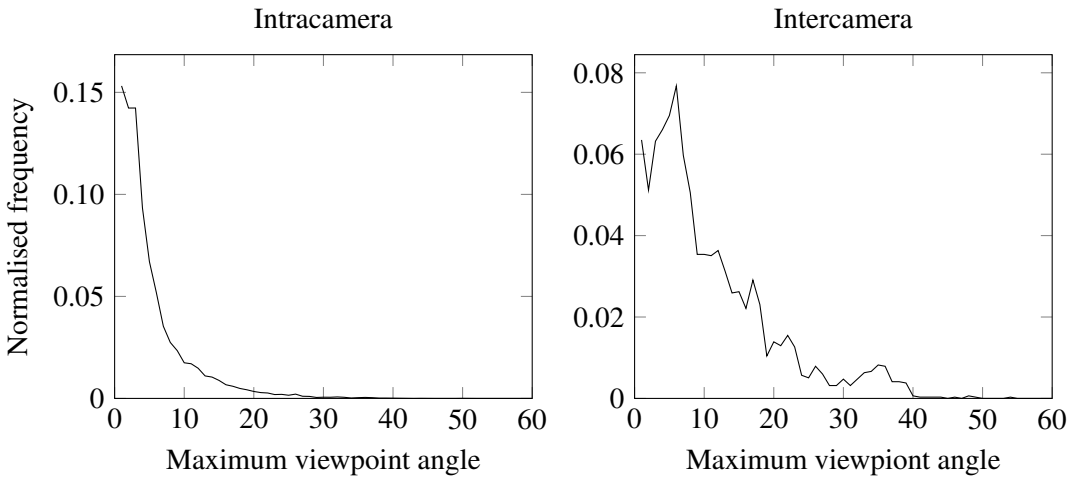


Figure 5.7: Distribution of the maximum viewpoint angle (in degrees) for MST features within each camera for all testing sequences (on the left, intracamera), and for correct matching MST features between cameras for all sequence pairs (on the right, intercamera). The maximum viewpoint angle corresponds to the angle of the pair of views that is maximum across all the possible view pairs where the 3D point corresponding to the spatio-temporal feature is visible. Note the different scale of the y-axis. The total number of estimated MST features is 50791. The total number of correct matches is 3165.

of the descriptor itself (*e.g.* ORB is not robust to viewpoint differences). The distribution of the maximum viewpoint angle for correctly matched MST features across cameras shows that the proposed approach can handle differences of up to 40 degrees.

Finally, Figure 5.8 shows the average F_1-scores and matching times, while Table 5.5 shows the total time to match the descriptors for each sequence pair. The total matching time depends on the number of feature trajectories in each sequence. Set representations such as SetDesc, MST-S, and MST have a higher F_1-score but, as expected, are slower than LMED, T-D, and T-DS, because of the *set2set min dist* strategy. The computational cost of MST-S and MST is quadratic with respect to the number of scales, $\mathcal{O}(S^2)$, whereas the computational cost of SetDesc depends on the length of the trajectories, thus resulting in a large (temporal) standard deviation. On average, the matching performance of these approaches is higher than LMED, T-D, and T-DS, but with a larger deviation. Note that the additional stability vector in T-DS and MST associated with the weighted Hamming distance (Eq. 3.17) doubles the matching time with respect to their counterparts, T-D and MST-S. As a reference for a histogram-based descriptor, we also report the total matching when applying SetDesc and T-D to SIFT. However, the timing between the two employed image-based descriptors are not comparable as the number of feature trajectories, as well as their length, differs from each other.

Table 5.5: Total matching time for each sequence pair and for each spatio-temporal feature (in seconds). Note that SIFT and ORB are not comparable due to the different number of feature trajectories for each sequence. Observe the comparison between spatio-temporal features for each row. Legend – Seq.: sequence; Desc.: descriptor; #FT: number of feature trajectories.

Seq. pair	**Desc.**	**#FT**		**Total matching time (s)**					
		Seq1	Seq2	SetDesc	LMED	T-D	T-DS	MST-S	MST
desk	SIFT	1486	658	46		10			
	ORB	1198	1005	55	13	14	14	16	23
office	SIFT	2402	726	45		17			
	ORB	2305	2566	115	55	56	64	71	104
courtyard	SIFT	3214	3830	939		151			
	ORB	4416	4950	331	186	188	216	241	358
gate-1	SIFT	2673	465	35		11			
	ORB	7806	7000	1288	492	486	558	625	904
gate-2	SIFT	2673	1763	186		47			
	ORB	7806	7185	1575	519	515	587	657	942
gate-3	SIFT	2673	1788	216		53			
	ORB	7806	7211	1633	522	519	596	664	947

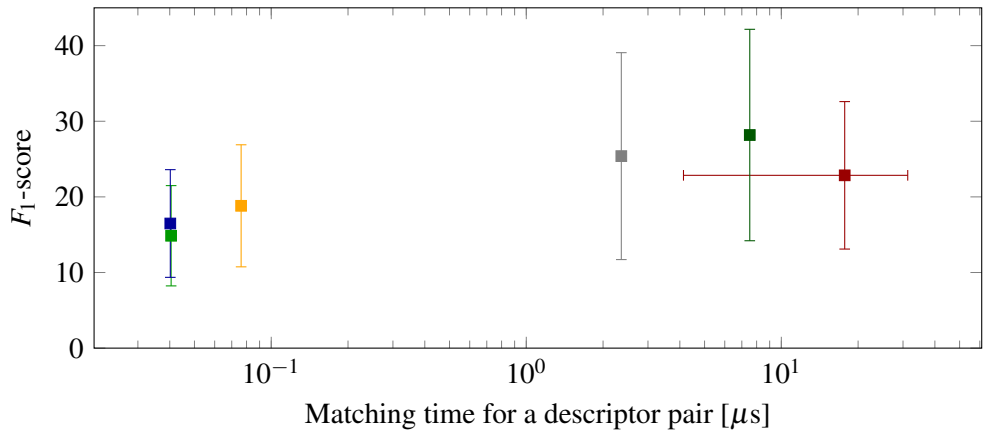


Figure 5.8: Accuracy and efficiency averaged across all the sequence pairs. The number of initial localised ORB features set to 2000. Note the large standard deviation in the efficiency for SetDesc due to the varying length of the sets. Legend: ■ LMED, ■ T-D, ■ T-DS, ■ MST-S, ■ MST, ■ SetDesc.

5.3.3 Comparison of multi-scale temporal feature with bag of visual words

Figure 5.9 compares the F_1-score of our proposed method, MST, against BoW, the cross-camera matching based on ORB-SLAM and the Bag of Visual Binary Words [33, 68]. We consider three variants of BoW: all the keyframes of camera 1 are compared against all the keyframes of the second camera 2 (BoW-A1A2); the last keyframe of camera 1 is compared against all the keyframes of camera 2 (BoW-L1A2); and the last keyframe of camera 2 is compared against all

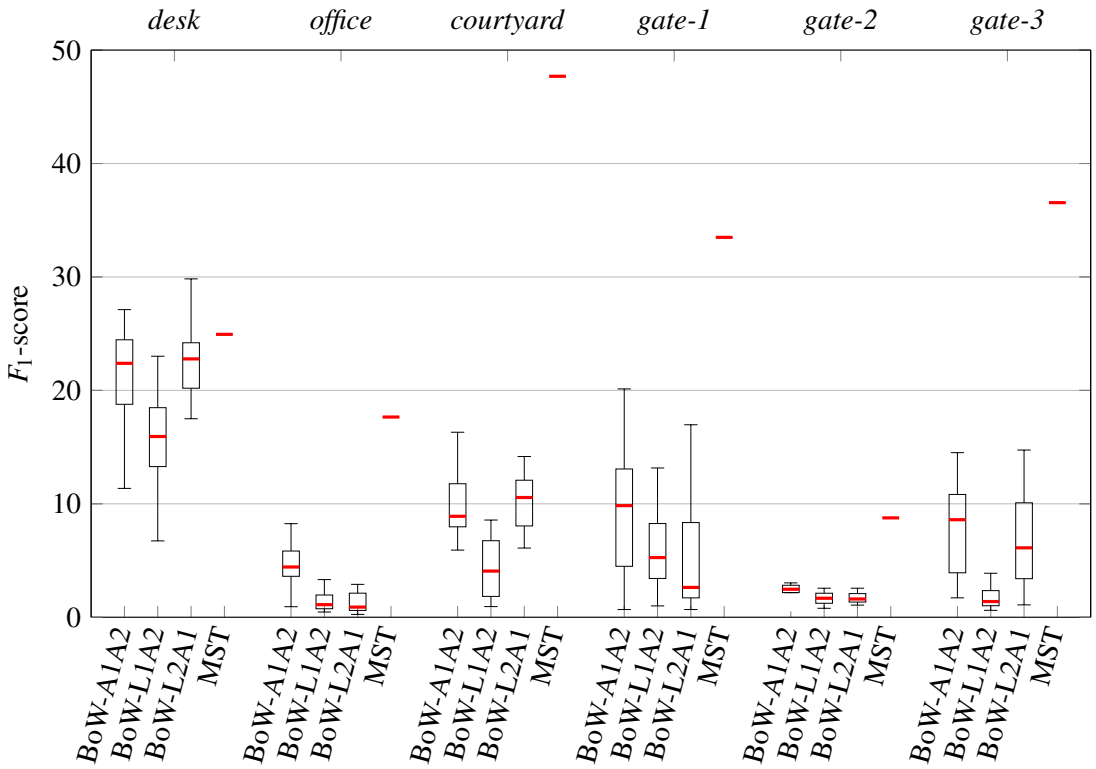


Figure 5.9: F_1-score comparison between MST and the BoW approach. For each sequence pair, we show three cases for BoW. BoW-A1A2: the best image match is estimated among all keyframes of both cameras. BoW-L2A1: the best image match is estimated between the last keyframe of the second camera against all the keyframes of the first camera; BoW-L1A2: the best image match is estimated between the last keyframe of the first camera against all the keyframes of the second camera. BoW-based matching results are obtained by running ORB-SLAM [68] over 30 runs for both camera sequences simultaneously.

the keyframes of camera 1 (BoW-L2A1). The last two variants recall scenarios where only one keyframe (usually the last) is sent/received by each camera [32, 81]. To account for the non-deterministic nature of ORB-SLAM, we run ORB-SLAM 30 times for each sequence using the same settings of our approach. While BoW creates a feature vector using all the local features of a frame, the matching within ORB-SLAM limits the valid matches to features with a corresponding 3D point, similar to our MST. MST outperforms BoW on all sequence pairs. In *desk* where geometric variations are small, MST slightly outperforms BoW, while the benefit of our approach is clearly visible in *courtyard*, *gate-1* and *gate-3* where geometric differences are more challenging. In the most severe viewpoint differences of *office* and *gate-2*, MST outperforms BoW by more than 10% and 5%, respectively. Note that in *gate-2* the two cameras approach the same point of the scene from different viewpoints.

5.3.4 Comparison of binary descriptors

The proposed spatio-temporal approaches are generic and can be applied to different image-based binary descriptors. As we model feature track extraction and spatio-temporal descriptor considering binary descriptors based on sampling patterns and dominant orientation, we analyse and compare the spatio-temporal approaches using BRIEF [17], ORB [77], LDB [114], and LATCH [51] as baselines. Note that we steer all binary reference descriptors according to the estimated orientation using the intensity centroid method [75]. We integrate the OpenCV implementation of BRIEF, ORB, and LATCH, and the author's implementation of LDB[6] in our own implementation.

While BRIEF and ORB compares intensity values of pixel pairs, LDB compares the mean intensity and the directional gradients of regular sub-windows within the patch with a multi-grid approach; and LATCH compares the norm of the difference between two sub-windows using a triplet of sampling points within the patch, with one point acting as anchor. It is noteworthy that most of the binary descriptors smooth the image (or scale level in an image pyramid) to reduce the sensitivity to noise in the intensity values [17, 77, 114], unless small windows are used (*e.g.* LATCH [51]).

We also include DeepBit [53] in the comparison, as a learnt CNN-based but non sampling-pattern based descriptor, and RFD [29] (both RFD_R and RFD_G), as a binary descriptor based on receptive fields followed by thresholding. Note that the dimensionality of previous descriptors is 256 bits, while the dimensionality of RFD_R is 293 and that of RFD_G is 405[7]. Unlike previous descriptors, DeepBit cannot directly be employed within the full method, such as feature point tracking, and therefore we applied DeepBit on the patches belonging to feature tracks extracted using ORB features. We consider the 256 bit version trained on the *Liberty* (DB-L), *NotreDame* (DB-N), and *Yosemite* (DB-Y) landmarks of the UBC Phototourism dataset [111]. To also compare with a histogram-based descriptor, we provide results of SetDesc and T-D applied to SIFT [55].

Figure 5.10 shows the F_1-score performance averaged across all sequence pairs using $F = 2000$. We can observe that RFD is a better choice for any of the spatio-temporal approaches given its higher accuracy, while DeepBit is the worst, followed by LATCH. The performance

[6] http://lbmedia.ece.ucsb.edu/research/binaryDescriptor/web_home/web_home/index.html, accessed: Dec 2018

[7] http://www.nlpr.ia.ac.cn/fanbin/rfd.htm

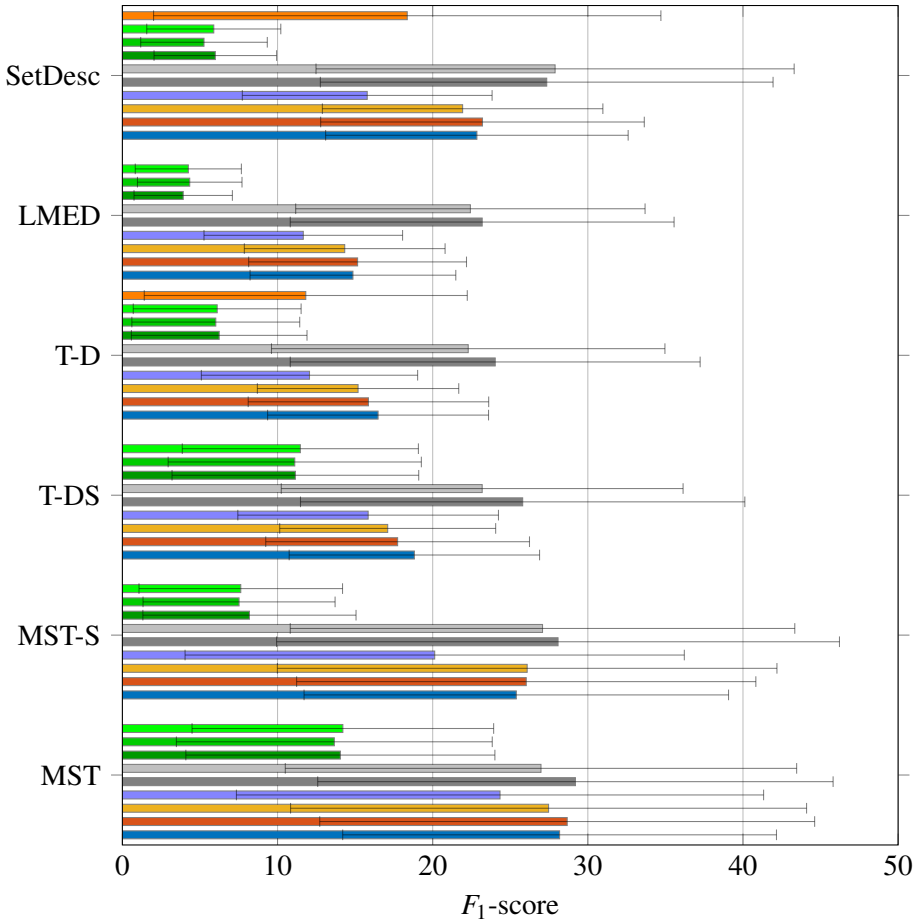


Figure 5.10: Average F_1-score and standard deviation across all sequence pairs, targeting a maximum of 2000 local features per frame during localisation. Comparison between binary (ORB), histogram-based (SIFT), and CNN-based (DeepBit) descriptors. Note that for SIFT, we compute only the set of SIFTs over time (SetDesc) and the average within the set as reduction (T-D). Legend: SIFT, DeepBit (Yosemite), DeepBit (Notre-Dame), DeepBit (Liberty), RFD_R, RFD_G, LATCH, LDB, BRIEF, ORB.

of LATCH and DeepBit shows how learning on a specific dataset (Phototourism) makes generalisation to other scenarios still a challenge. When using our multi-scale approach, MST, ORB, LATCH, and LDB become other valid alternatives to RFD. Note that selecting stable bits marginally improves the average performance of MST over MST-S.

Table 5.6 compares the timings[8] to extract the spatio-temporal features when employing the different image-based descriptor on three testing sequences (*desk*, *courtyard*, and *gate-1*) with varying image resolutions and content. We compare the impact of SIFT, ORB, BRIEF, LDB, LATCH, RFD on the overall extraction of the (multi-scale) spatio-temporal features in terms of detection time per image feature, the multi-scale description time per image feature, the tracking

[8] All the experiments are performed using a machine with Intel ®Core i7-4790S CPU @ 3.20GHz × 8, 15.6 GBi RAM, and running Ubuntu 18.04

Table 5.6: Efficiency analysis on three testing sequences with different resolutions. Note that SIFT is not described at multiple scales. Legend – DET: detection time per feature; DESC: (multi-scale) description time per feature; TRACK: tracking time per feature; FRAME: average time to process a frame; VAL: timing for 3D geometric tests; RED: timing for temporal reduction; #FT: number of feature trajectories.

		Frame timings				**Post-processing timings**			
		DET (μs/feat)	**DESC** (μs/feat)	**TRACK** (μs/feat)	**FRAME** (s)	**#FT** before	**VAL** (s)	**RED** (s)	**#FT** after
desk	SIFT*	11.75	25.98	50.95	0.04	3431	1.38	0.01	1486
	ORB	16.82	60.63	842.73	0.42	1751	1.08	1.19	1198
	BRIEF	10.17	173.92	988.33	0.21	514	0.49	0.50	423
	LDB	10.91	177.04	889.67	0.16	638	0.33	0.45	486
	LATCH	9.51	301.16	932.19	0.16	755	0.25	0.41	525
	RFD_R	10.12	1645.10	2492.61	1.25	4159	0.95	1.11	992
	RFD_G	9.90	14474.67	15066.39	7.46	5215	0.49	1.02	1055
courtyard	SIFT*	15.61	16.79	37.39	0.07	7001	3.61	0.03	3214
	ORB	11.73	30.90	318.05	0.37	12055	0.56	2.00	4416
	BRIEF	9.10	67.63	391.47	0.20	3751	0.31	1.10	2225
	LDB	8.74	111.65	386.26	0.15	4716	0.18	0.75	1887
	LATCH	9.18	257.21	494.12	0.17	5782	0.09	0.42	1262
	RFD_R	23.36	1642.86	1946.34	1.86	15512	0.34	1.18	2346
	RFD_G	23.82	14546.47	14463.49	13.93	17512	0.17	0.78	1624
gate	SIFT*	13.12	28.41	57.01	0.10	8373	11.22	0.02	2673
	ORB	13.36	30.54	726.64	1.09	19713	6.38	6.64	7806
	BRIEF	8.91	45.59	1248.11	1.86	12035	14.69	8.08	6316
	LDB	9.33	74.26	737.62	0.91	15365	4.89	5.60	6818
	LATCH	8.82	202.29	726.02	0.80	18403	2.86	4.54	6438
	RFD_R	23.83	1623.85	2293.06	2.64	27017	3.71	4.02	4560
	RFD_G	23.88	14429.77	14690.71	15.61	31392	1.41	2.84	3632

time per image feature (including both Kanade-Lucas-Tomasi tracker, descriptor extraction, and descriptor validation), the average time per frame, and the post-processing time consisting of the 3D geometric tests and temporal reductions. To make the comparison fair, all the binary descriptors are integrated within the same implementation, except DeepBit that extracts the descriptors from the patches of the final feature trajectories obtained with ORB. We refer the reader to the analysis of the running times provided by [53], which shows that the processing of the patches in batches makes the extraction slow and not comparable with other binary descriptors in our application. As SIFT is also integrated and adapted in the framework, we report its results as reference.

We can observe that even though RFD is the most accurate in Figure 5.10, the average frame

processing time is highly affected, especially due to the high extraction time of the descriptor at multiple scales[9]. Even if ORB is the fastest among the sampling-pattern based approaches in describing each feature at multiple scales, LDB and LATCH require less time, on average, to process each frame. The single scale extraction of the SIFT descriptor achieves the fastest processing of a frame, on average. Then, it is important to note how each image-based descriptor affects the number of estimated feature trajectories before the 3D geometric tests and temporal reduction, and that this number largely varies. Moreover, each feature trajectory varies in length affecting the final timing to validate in 3D, which is more noticeable in sequence 1 of *gate* that contains 100 frames instead of 50 and has a higher resolution (1280×720 pixels). Note that ORB has the highest number of feature trajectories after the 3D geometric tests across all the sequences, except *desk* where SIFT obtains a higher number. The temporal reduction is done for all validated trajectories and for LMED, T-D, T-DS, MST-S, and MST, all in once. Again, this timing is affected by both the number of feature trajectories and their varying length. SIFT is the fastest because the temporal reduction is performed only for T-D.

5.4 Cross-camera place recognition

5.4.1 Experimental setup

To validate XC-PR, we consider two variants for ATST: binary search tree (*BTST*) and ternary search tree (*TTST*); and two alternatives: an incremental list of TWs (*LiST*) and the adapted loop closure detection approach with the frame-based DBoW [33, 69]. LiST aims to reproduce BoTW [102] with binary features, but without guided matching and feature point detection for each frame. LiST is also adaptive as ATST.

DBoW [33, 69] describes each frame with binary features stored in a vocabulary tree, which is trained offline. To adapt DBoW to XC-PR, we remove the temporal consistency check as this condition cannot be applied across cameras. Since ATST and LiST do not share data every frame, we share binary descriptors for the adapted DBoW at regular intervals, *e.g.* every 5 frames, similarly to the observed average rate for localising new binary features in XC-PR. Because of the pre-trained vocabulary and the need to share all the descriptors for the geometric verification, DBoW reconstructs the BoW vector at the receiving camera to find the most similar image.

For all methods, we extract $F = 1000$ ORB descriptors [77] at a single scale with a thresh-

[9] Slow extraction time was also observed in [13].

Table 5.7: Parameter settings for DBoW, LiST and ATST (BTST and TTST). *The initialisation of LiST and ATST is not fixed (see Eq. 4.3).

Parameter		**DBoW**	**LiST**	**ATST**
Max. # of features/frame	F	1000	1000	1000
Initialisation (min. # of frames)	η	30	*	*
Ratio for min. # active TWs/frame	χ	–	0.6	0.6
Min. length of TWs	ρ	–	10	10
Max. # of frames for view selection	Λ	–	50	50
Max. # of TWs/node	N	–	–	50
Threshold for Lowe's ratio test	δ	0.8	0.8	0.8
Max. Hamming distance	γ	50	50	50

old of the FAST detector set to 25 [76], and with a grid-based suppression to favour a spatially uniform distribution (see Chapter 3.4). We track interest points using the OpenCV implementation of the pyramidal Kanade-Lucas-Tomasi tracker with a window size of 21×21 pixels and 5 scales [15, 87]. The minimum length of a TW is $\rho = 10$ frames to limit the growing number of TWs. To trigger a new localisation, we set $\chi = 0.6$. For view selection, we set the maximum number of frames within the adaptive temporal window to $\Lambda = 50$. When matching TWs, we set the maximum Hamming distance to $\gamma = 50$ as threshold to validate a match (typical separation of matching and non-matching feature distributions in the space of the Hamming distances for binary descriptors with $D = 256$ [17, 68]), while the threshold for Lowe's ratio test (or nearest neighbour distance ratio, NNDR) [55] to $\delta = 0.8$. For BTST and TTST, we initially set the maximum number of TWs for each node to $N = 50$, similarly to HBST [79], but we analyse the performance of both methods when varying N. Moreover, we set the minimum number of frames before sharing visual words for DBoW to $\eta = 30$. This allows DBoW to populate an initial database of BoW to be comparable to the ρ-dependant initialisation period of LiST, BTST, and TTST. Table 5.7 lists the value of the parameters used in the experiments.

We implement the decentralised approach using ZeroMQ[10] distributed messaging with the request-reply strategy, *i.e.* after sending a message, the camera waits for a reply from another camera before processing the new frame, affecting the speed. As the frame processing and place recognition are performed in parallel in two different threads, the two cameras operate asynchronously, and their start and end times might differ. This means more places are included in the validation.

We use pairs of sequences of different scenarios captured with multiple hand-held cam-

[10] http://zeromq.org/

eras, from publicly available datasets: TUM-RGB-D SLAM [94]; the scenario *courtyard*[11] from CoSLAM [121]; and sequences we collected and make available to the research community. From TUM-RGB-D SLAM we use the sequences fr1_desk, fr1_desk2, and fr1_room to form the *office* scenario. The scenario *courtyard* consists of 4 sequences whose length varies between 3 to 4 mins acquired around an outdoor university courtyard area, starting and ending approximately at the same positions. For annotation purposes, we sub-sample the sequences from 50 to 25 fps. For our own dataset, we collected the *gate* and *backyard* scenarios, each consisting of 4 sequences (1280×720 pixels) at 30 fps with varying duration, in different outdoor scenarios with both hand-held and chest-mounted cameras. As for *courtyard*, we sub-sample *backyard* from 30 to 10 fps due to the high number of frames and for annotation purposes. All the sequences together results in a total of ∼28,000 frames and a duration of approximately 25 mins. Table A.1 summarises the characteristics of each scenario in terms of number of frames, duration, frame-rate, resolution, and platform. See Appendix A for details about the dataset and its annotation. For simplicity, we refer to sequence pairs with an abbreviation, *e.g. gate 1|2* as *G1|2*, or *courtyard 2|4* as *C2|4* in the rest of the section.

As performance measures, we compute place recognition accuracy and speed to evaluate the methods. *Place recognition accuracy* assesses the capability of a method to correctly recognise a previously seen place in a camera for each query from another camera, compared to the annotation provided with the dataset. Similarly to the evaluation of loop closure detection, we compute precision, recall, and F_1-score. *Precision* is the number of correctly recognised places over the total number of recognised places. *Recall* is the number of correctly recognised places over the total number of annotated view correspondences. *F_1-score* is the harmonic mean between precision and recall. Because of the pairwise approach, we compute precision and recall for each camera and we then average the F_1-scores between the two cameras. We use the average F_1-score to compare the methods. Moreover, we quantify the average speed of frame processing and place recognition for all methods, and feature tracking for LiST and ATST.

5.4.2 Results

Figure 5.11 compares precision and recall of BTST, TTST, LiST, and DBoW when varying the visual overlap threshold up to 75% on the sequence pair *G1|4* as example. All methods obtain maximum precision when the threshold is lower than 30%, while curves tend to decrease for one

[11] `http://drone.sjtu.edu.cn/dpzou/project/coslam.php`, accessed: March 2018

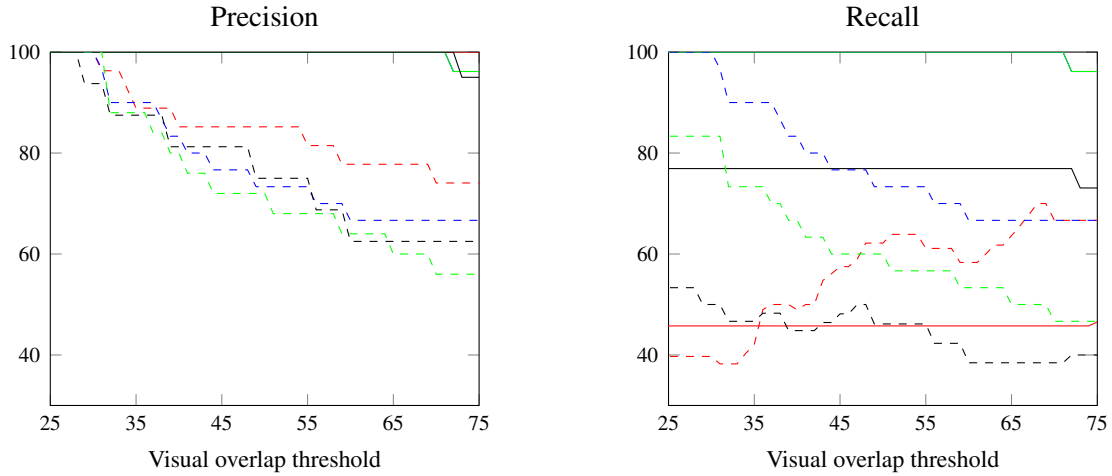
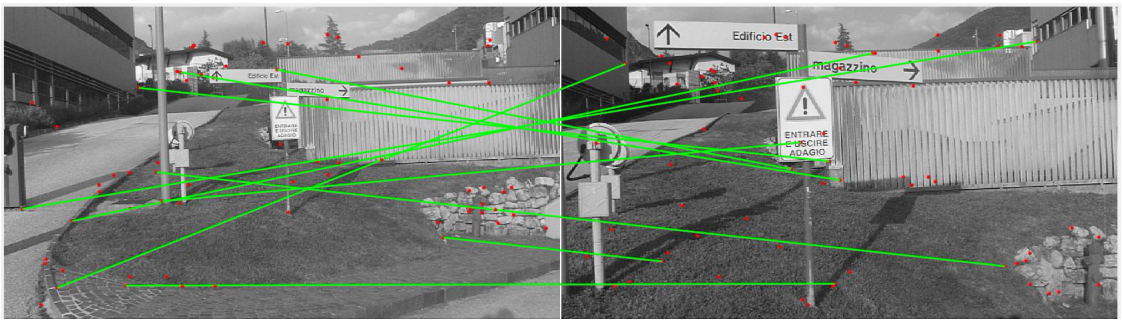


Figure 5.11: Precision and recall when varying the threshold of the visual overlap (percentage) on *gate 1|4* for a camera (solid lines) and the other camera (dashed lines). Image pairs are a valid correspondence if their visual overlap is greater than the threshold. Legend: — DBoW, — LiST — BTST — TTST.

camera, reaching 60% precision when the threshold is at 75%. Precision for LiST, BTST, and TTST decreases in similar way, while DBoW maintains higher precision. Also for recall, the results for one camera are stable, with LiST and BTST at 100%, while DBoW is the worst at 45.76%. For the second camera, LiST outperforms the other methods, while BTST is the second best performing. As for precision, recall curves decrease of about 30% for LiST and BTST, and of about 10% for TTST when increasing the threshold from 30% to 75%. Recall of DBoW, instead, increases of about 30%, achieving similar performance of LiST at 75%.

Table 5.8 compares the place recognition accuracy of all methods on *G1|2*, *G1|4*, and *O1|2* with different matching strategies to reduce ambiguities and avoid erroneous matches. We consider a matching strategy with no threshold, while we use $\gamma \in \{30, 50\}$ as maximum Hamming distance for the strategies based only on the threshold. For LiST and BTST, we also consider a variant where the stability vector is not used during matching (LiST* and BTST*). For LiST, BTST, and TTST, weevaluate the proposed dynamic threshold using the stability vector and fixing $\gamma = 50$ (see Eq. 4.2). For NNDR [55], we consider $\delta \in \{0.6, 0.8\}$. When $\delta = 0.6$, the matching strategy is more restrictive, enforcing a larger distance between the first two closest neighbours and resulting in fewer matches; whereas possible erroneous matches can be present when $\delta = 0.8$. The last strategy is the proposed one (see Chapter 4.3) that combines NNDR with $\delta = 0.8$ with the dynamic threshold for LiST, BTST, and TTST, or $\gamma = 50$ for DBoW, LiST*, and BTST*. When no threshold is used, all methods achieve high F_1-score, meaning that places



(92.41%, 4.04°, 3.94)

Figure 5.12: Example of correctly recognised place but with wrong matches (— inliers) between interest points (■) of corresponding tracked words in *gate* 1|4 (BTST with fixed threshold $\gamma = 50$). Note in bracket the high visual overlap, low angular distance, but large Euclidean distance (different scale).

are correctly recognised when validating the geometric model despite the presence of mostly outliers (see Figure 5.12). When using $\gamma = 30$ or NNDR, accuracy decreases for all methods, as expected. Moreover, the accuracy of LiST* and BTST* is worse than LiST and BTST, showing how the stability information benefits for recognising correct places across cameras. We can also observe how the dynamic threshold balances between a restrictive threshold ($\gamma = 30$) or a permissive threshold ($\gamma = 50$). The combination of the dynamic threshold with NNRD allows to remove further ambiguities, reducing the accuracy compared to only thresholding or improving the accuracy compared to use only NNRD. For example, BTST achieved 58.92% F_1-score with the dynamic threshold, 28.85% with NNRD ($\delta = 0.8$), and 30.99% with the combination of the two strategies on *O1|2*. Using the combined strategy, it is worthy to note that LiST achieves the highest accuracy in *G1|2* and *G1|4*, BTST outperforms TTST of almost twice F_1-score on all sequence pairs, and DBoW achieves the highest accuracy in *O1|2*. For the rest of the experiments, we discard LiST* and BTST* and we use only the proposed combined strategy for all methods.

Figure 5.13 shows the accuracy of BTST and TTST when varying the maximum number of stable TWs per node, N. While fixing the visual overlap threshold at 20%, we consider the following scheduling, $N \in \{25, 50, 75, 100, 150, 200, 250\}$, on some sequence pairs for each scenario. As it is not possible to reproduce the exact results for each method due to asynchronous exchange of data between cameras, we perform 5 runs for each N and report the average F_1-score and standard deviation. When N increases, both BTST and TTST can find more matches, which results in an increase place recognition accuracy with small standard deviation across runs for most of the cases. BTST achieves 81.81%, 93.95%, 72.04%, and 67.82% F_1-score for *G1|2*, *G1|4*, *B1|4*, and *B2|3*, respectively. In *G1|4*, *O1|2*, and *O1|3*, the accuracy of BTST be-

Table 5.8: Comparison of the place recognition accuracy (F_1-score) on three testing sequence pairs with different strategies to remove outliers when matching descriptors: no threshold, fixed threshold ($\gamma = 30$ and $\gamma = 50$), dynamic threshold (Dyn), nearest neighbour distance ratio (NNDR) with values 0.6 and 0.8, and the combination (Comb) of NNDR (0.8) with dynamic threshold. For DBoW, comb uses NNDR (0.8) and fixed threshold ($\gamma = 50$). Note that LIST* and BTST* do not use the stability information when matching tracked words.

Sequence pair	**Method**	**Matching strategies**						
		Threshold				**NNDR**		**Comb**
		No	30	50	Dyn	0.6	0.8	
gate 1|2	DBoW	92.50	92.67	93.67	–	91.65	94.97	92.50
	LiST*	100.00	25.53	95.84	–	23.08	47.82	79.90
	LiST	100.00	82.37	100.00	100.00	36.36	76.29	96.94
	BTST*	100.00	12.50	37.67	–	15.66	45.74	24.73
	BTST	100.00	26.32	100.00	84.35	17.65	79.00	61.89
	TTST	100.00	22.22	100.00	55.69	14.63	69.26	37.04
gate 1|4	DBoW	62.08	60.56	62.08	–	59.66	63.10	58.27
	LiST*	100.00	13.33	95.37	–	13.33	28.94	78.41
	LiST	100.00	85.83	100.00	100.00	23.53	80.62	98.28
	BTST*	98.28	3.70	23.53	–	13.33	58.33	23.53
	BTST	98.28	18.75	95.61	80.85	16.13	84.44	59.66
	TTST	100.00	10.34	99.02	50.85	13.33	92.44	46.61
office 1|2	DBoW	81.88	80.70	78.55	–	79.08	75.27	78.03
	LiST*	69.62	22.46	58.34	–	9.05	24.53	33.15
	LiST	71.27	40.34	70.63	67.80	16.60	35.05	38.22
	BTST*	66.92	3.49	45.25	–	.45	16.90	22.15
	BTST	69.70	34.11	65.89	58.92	6.36	27.85	30.99
	TTST	62.12	9.05	54.39	40.87	.00	18.70	19.71

comes comparable to LiST, showing the advantage of using the proposed hierarchical structure to achieve higher speed at similar accuracy (see also Figure 5.14). For challenging sequence pairs, such as *G2|3* or in *office* and *courtyard*, the accuracy of BTST, TTST, and LiST is lower than 50% F_1-score. Moreover, TTST achieves similar accuracy of BTST on *G1|2*, *G2|3*, *C1|2*, and *C1|3* when $N = 250$. The improvement from $N = 25$ to $N = 250$ for both BTST and TTST is higher than 40% F_1-score in *G1|2* and *G1|4*, and between 10% and 30% F_1-score for other sequence pairs due to different degrees of geometric differences in each scenario.

Table 5.9 shows the complete evaluation of the place recognition accuracy averaged over 5 runs for BTST and TTST ($N = 250$) compared to DBoW and LiST on all sequence pairs. We also include the comparison among all the four methods on the first 500 frames for all sequence pairs in *courtyard*. BTST achieves high accuracy (F_1-score $> 75\%$) in *G1|2* and *G1|4* due to the limited viewpoint differences, and outperforms DBoW on *B1|4* and *B2|3*. We do not report the results

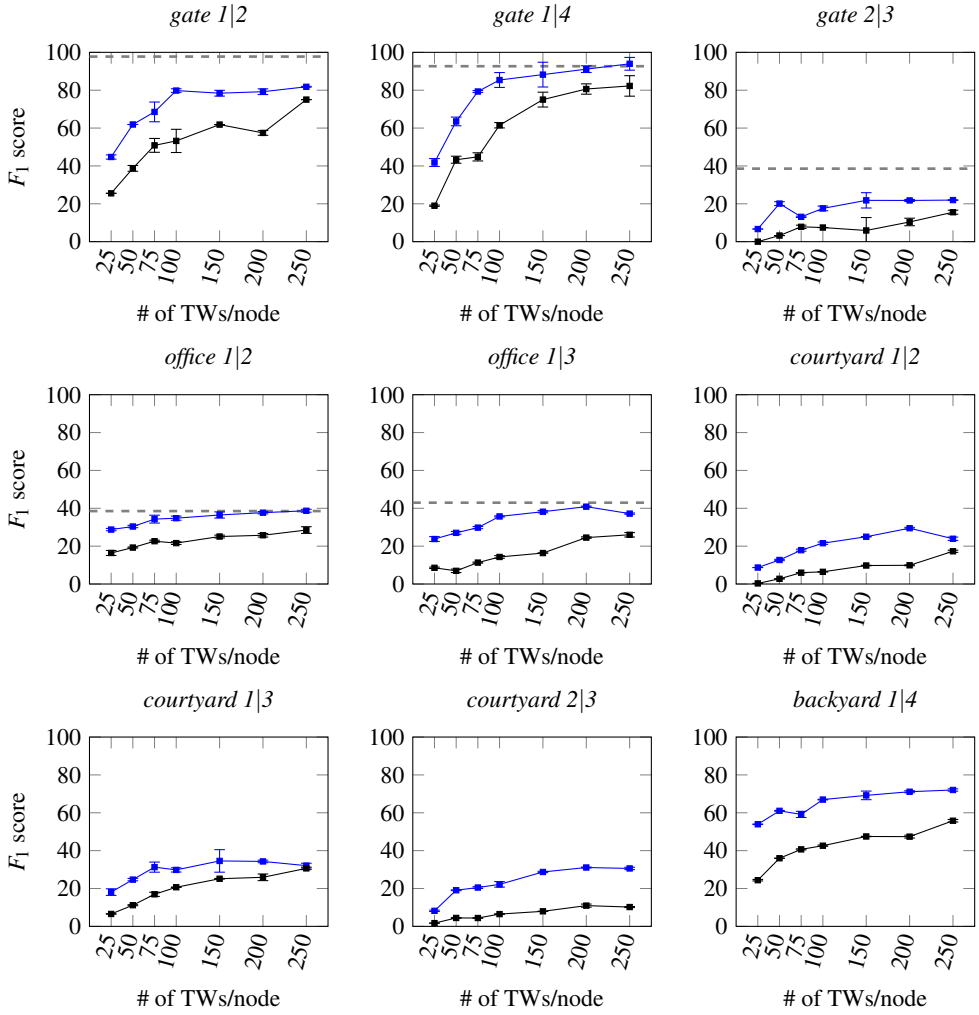


Figure 5.13: Average F_1 score and standard deviation over 5 runs when varying the maximum number of tracked words per node (# of TWs/node) using BTST (—) and TTST (—). Note that we report F_1-score for LiST as reference (dashed black line) on *gate 1|2*, *gate 1|4*, *gate 2|3*, *office 1|2*, and *office 1|3*, while LiST was not run on the other sequences due to the high computational time (see Figure 4.5).

of the other sequence pairs in *backyard* as the accuracy is lower than 10% due to the challenging viewpoint differences. *B1|4* and *B2|3* contain sequences acquired with a chest-mounted and a hand-held camera from the same walking person, resulting in limited viewpoints differences. On *courtyard*, BTST achieves comparable accuracy with DBoW (about 20% and 30% F_1-score on average) on sequence pairs *C1|3* and *C1|4*, while accuracy is half F_1-score for *C2|3* (30.60%), *C2|4* (22.26%), and *C3|4* (28.27%). Overall, all methods obtain a standard deviation lower than 2% F_1-score, except for the scenario *gate* where variations can be higher, *e.g.* up to 6.36% for DBoW (*G1|3*) and 16.40% for LiST (*G1|4*). These variations are mainly resulting from different recall values across runs with precision often at 100%; however, precision lower than 100% in some runs affects the final F_1-score. For example, LiST achieved 15% recall in one camera for

Table 5.9: Comparisons of F_1-score results averaged over 5 runs for each sequence pair of all testing scenarios, when the threshold on overlap ratio of the visual hull is 20%. Note also the results for each sequence pair in courtyard using the first 500 frames. Standard deviations in brackets.

Scenario	**Pair**	**Method**							
		DBoW		**LiST**		**BTST**		**TTST**	
gate	1|2	93.85	(1.13)	97.76	(1.33)	81.81	(.22)	75.04	(.00)
	1|3	68.56	(6.63)	80.31	(2.41)	65.90	(.65)	45.59	(.00)
	1|4	61.77	(1.56)	92.67	(16.40)	93.95	(3.37)	82.26	(5.41)
	2|3	58.95	(1.51)	38.57	(.69)	21.96	(.13)	15.54	(1.06)
	2|4	38.61	(1.26)	72.19	(.26)	29.68	(3.13)	18.92	(1.08)
	3|4	31.87	(4.32)	54.66	(2.08)	23.54	(.00)	11.24	(1.50)
office	1|2	78.78	(1.37)	38.52	(2.79)	38.70	(.76)	28.51	(1.77)
	1|3	47.13	(2.68)	42.96	(1.48)	37.19	(.34)	26.01	(1.17)
	2|3	49.71	(1.61)	26.05	(1.83)	14.98	(.94)	5.80	(.44)
backyard	1|4	47.35	(3.97)	–		72.04	(.61)	55.80	(.65)
	2|3	54.36	(3.20)	–		67.82	(.67)	49.09	(1.52)
courtyard	1|2	31.76	(.93)	–		23.91	(.87)	17.29	(.57)
	1|3	30.82	(1.43)	–		32.08	(1.26)	30.63	(.65)
	1|4	22.84	(1.28)	–		17.15	(.11)	5.95	(1.04)
	2|3	57.86	(1.81)	–		30.60	(.62)	10.20	(.27)
	2|4	34.11	(1.55)	–		22.26	(.71)	11.35	(.44)
	3|4	43.09	(.89)	–		28.27	(.54)	6.91	(.38)
courtyard (500)	1|2	55.60	(4.05)	73.18	(.44)	51.72	(2.15)	42.80	(1.72)
	1|3	62.53	(.47)	61.45	(.73)	31.25	(2.48)	31.32	(1.51)
	1|4	46.27	(2.75)	68.08	(.43)	59.09	(.86)	32.77	(7.09)
	2|3	70.32	(3.43)	69.95	(.46)	42.25	(1.23)	17.02	(.24)
	2|4	61.06	(1.10)	67.39	(.24)	44.18	(.43)	35.71	(.42)
	3|4	89.42	(1.27)	77.05	(3.03)	82.43	(1.93)	19.24	(.00)

one run and 100% for other runs in *G1|4*; or BTST obtained 66.67% precision and 7.65% recall for some runs in *G1|3*. It is also worth noting that the accuracy of TTST is lower than BTST in all sequence pairs. When limiting the sequences in *courtyard* to the first 500 frames, LiST and DBoW achieve the highest accuracy between 60% and 80% F_1-score, with DBoW being the best in *C3|4* (89.42% F_1-score).

Finally, we compare in Figure 5.14 the average speed of XC-PR when using BTST, TTST, LiST, or DBoW across all frames and cameras on one sequence pair, *e.g. G1|2*. Note that the speed for place recognition is averaged only across frames that request to perform place recognition in another camera. The lack of temporal information makes DBoW the fastest in terms of frame processing and place recognition, but at the cost of less informative features. As expected, the hierarchical organisation of the stable TWs allows BTST and TTST to be faster than LiST

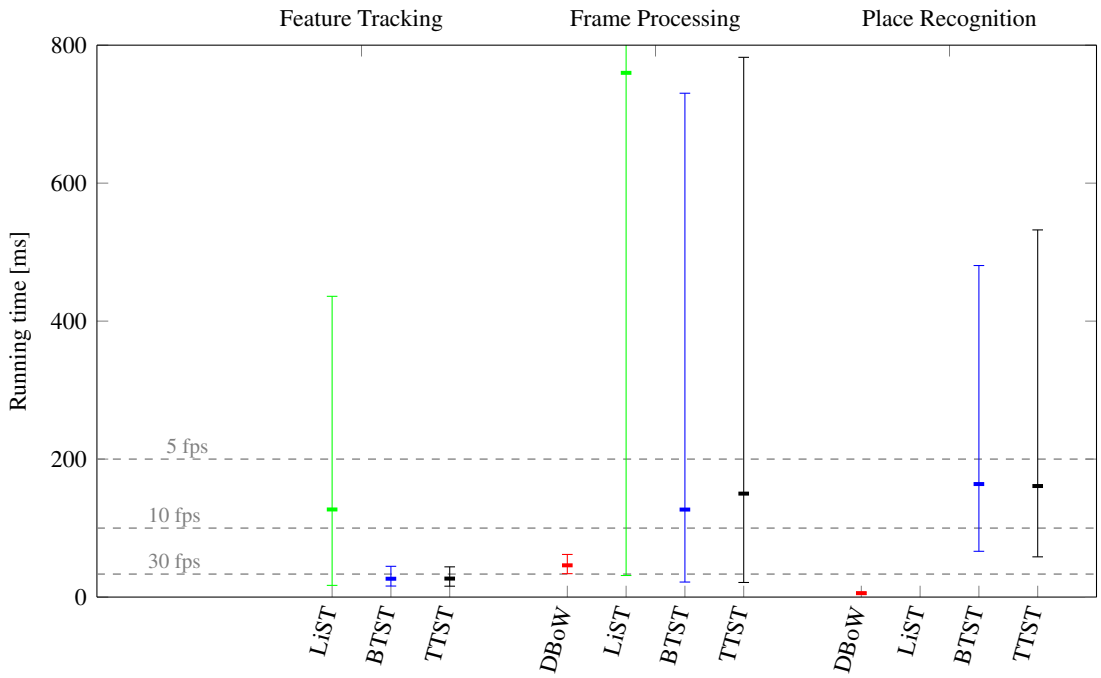


Figure 5.14: Comparison of the average speeds across frames and cameras on *gate* 1|2 (one run). We also report 5th and 95th percentiles. Note that we limit y-axis to 800ms, as the processing is near 1 frame/sec (fps). The average place recognition time for LiST is about 4.5 seconds. Legend: — DBoW, — LiST, — BTST, — TTST.

in frame processing and place recognition, on average. BTST and TTST run place recognition at 5-10 fps, on average. LiST, BTST, and TTST can process frames up to 30 fps, except for frames where the camera updates the tree, localises new binary features, and waits for the reply of the other camera after performing place recognition, in the current implementation. Indeed, processing these frames is the most time-consuming operation for BTST and TTST, on average between 400ms and 600ms. However, a number of TWs to share that is lower than the minimum allowed for the geometric validation avoids XC-PR to perform place recognition in the other camera and wait for a reply, or a number of matched Tws that is lower than the same minimum makes terminate the place recognition earlier. These conditions therefore cause large variations in speed for processing corresponding frames.

5.5 Summary

In this chapter, we evaluated the proposed descriptors, namely the multi-scale binary descriptor (MORB), the spatio-temporal descriptors T-D and T-DS, and the multi-scale temporal descriptor (MST); and the proposed cross-camera place recognition (XC-PR) approach.

For the image matching problem, we showed that MORB outperforms other binary descriptors, whose extraction is performed at the detection scale, under increasing scale difference between image pairs, and under different geometric and photometric variations. While the proposed multi-scale descriptor is based on ORB [77], the overall pipeline is modular and can be generalised to other (binary) interest point descriptors.

We then proposed a novel procedure for evaluating the matching of local spatio-temporal features in short image sequences acquired by independently moving cameras. The proposed evaluation is based on projective geometry [38] and extends the evaluation of local image features for image matching. We showed that the compact representation based on the most frequent and most stable binary values over time, T-DS, increases the number of correct matches compared to a high-dimensional descriptor represented by the set of binary descriptors, and the reduction strategy proposed within ORB-SLAM [68].

We also proved that MST outperforms all the previous spatio-temporal features and, even though the efficiency is reduced due to the high-dimensional multi-scale descriptor, MST improves the matching performance with respect to the efficient Bag of Binary Words approach, when validating the features through local 3D reconstruction. Moreover, we showed that the proposed spatio-temporal features are generic for a range of (image-based) binary descriptors.

Last, we validate XC-PR on annotated multi-camera scenarios, showing that organising compact spatio-temporal descriptors in an adaptive hierarchical structure (ATST), *e.g.* a binary search tree, allows to achieve similar accuracy but at higher speed than using an incremental list of tracked words. Moreover, the accuracy of XC-PR when using ATST is comparable to the accuracy of XC-PR when using the frame-based bag of binary words (DBoW [33]), while retaining only informative local features.

Chapter 6

Conclusion

6.1 Summary of achievements

In this thesis, we addressed two main problems regarding the view matching between cameras that move independently within unknown environments. The first problem involved the design of compact local features that can cope with severe scale and/or viewpoint differences. We presented three local features with binary descriptors, namely MORB, T-DS, and MST, and corresponding dissimilarity measures for view matching under scale and viewpoint differences, when cameras move independently in non-planar scenes (Chapter 3). The second problem involved the visual recognition of previously seen portions of a scene across two moving cameras. We introduced XC-PR, a novel cross-camera place recognition approach that identifies previously seen places across cameras, while exchanging over time selected compact spatio-temporal descriptors extracted locally for each camera. To efficiently search and match query descriptors, XC-PR organises for each camera the descriptors in a hierarchical structure that is updated while the camera moves and named Adaptive Tree of Stable Tracked words (ATST) (Chapter 4). In addition to the design of the proposed features and the proposed XC-PR framework, we summarise other achievements that we obtained in this thesis.

MORB is a binary descriptor that uses multiple scales of a Gaussian pyramid to increase the matching accuracy under scale changes. We also proposed a scale-aware neighbour matching strategy that estimates the minimum cross-scale distance between two MORB descriptors and, as a by-product, can infer the scale ratio between pairs of local features. We showed that the

proposed multi-scale descriptor, MORB, outperforms other local image-based binary features (*e.g.* LATCH [51], ORB [77], and their variants) whose descriptors are extracted at the detection scale, but the efficiency is reduced due to the feature matching process that compares all the scales between descriptor pairs. We also showed that the matched scales tend to differ from the scales where the interest points were localised, leading to an increase in the number of correct matches. While the proposed feature is based on ORB [77], the overall pipeline is modular and can be generalised to other (binary) local image descriptors.

Moreover, we observed that, when using the nearest neighbour strategy and by varying the threshold to determine valid matches, the area under the recall vs 1-precision curve does not preserve the ranking consistency with respect to the area under the precision curve and the area under the recall curve for comparing the approaches. Therefore, we proposed to compute, as performance measure, the nearest neighbour average F_1 score, which computes the area under the F_1 score curve and preserves the ranking consistency.

T-DS is a spatio-temporal binary descriptor obtained by tracking ORB [77] features and concatenating their descriptors for investigating the problem of matching spatio-temporal features extracted from videos acquired by independently moving cameras. As matching the high-dimensional descriptors is computationally expensive, we accumulated the spatio-temporal features into fixed-length binary descriptors, by pooling and selecting the temporally dominant values. We also complemented this descriptor with an additional binary descriptor that encodes the temporal stability of each binary test, and is used to ignore binary values in the first descriptor at corresponding element locations, when matching features across cameras.

MST is a novel multi-scale temporal binary descriptor that accounts for both viewpoint and scale variations across moving cameras in non-planar scenes. The proposed descriptor encodes, at multiple scales, temporal dominant and stable binary values of the neighbourhood of a 3D point, as observed in the image sequence of a camera and obtained by tracking the corresponding image-based binary feature. The similarity matching strategy uses a scale-aware weighted Hamming distance to handle scale variations and to account for the instability of the binary values. Experiments showed the advantage of the proposed approach, in terms of accuracy, over alternative approaches for reducing the temporal descriptors and the bag of visual word approach between two cameras performing visual SLAM. In terms of efficiency, MST outperforms a simple approach without reduction; however, the matching of these multi-scale temporal features is still

computationally expensive due to the cross-scale distance computation. We also showed that the framework is generic for a range of binary features, such as BRIEF [17], ORB [77], LDB [114], LATCH [51], and DeepBit [53]. In addition to this, we proposed a procedure to annotate reference correspondences using multi-view geometry [38], and to evaluate spatio-temporal features.

Finally, we presented XC-PR, a novel Cross-Camera Place Recognition approach that, for each camera, locally describes and reduces tracked binary features into compact and informative spatio-temporal descriptors, encoding the most persistent values over time. At automatically selected frames, each camera independently exchanges selected descriptors to recognise previously seen places in another camera. To efficiently search and match query descriptors, each camera independently organises the descriptors in an Adaptive Tree of Stable Tracked words (ATST) that is selectively updated while the camera moves. XC-PR recognises a place by geometrically validating the previous frame with the highest number of binary features corresponding to matched descriptors. We demonstrated the proposed XC-PR on four different scenarios with multiple cameras moving independently in non-planar scenes. Experiments show that using ATST with a binary search tree allows XC-PR similar cross-camera place recognition accuracy but faster, on average, than a baseline consisting of an incremental list of spatio-temporal descriptors. Moreover, XC-PR with ATST achieves similar accuracy of a frame-based bag of binary words approach (*e.g.* DBoW [33]) adapted to our approach, while avoiding to match features that cannot be informative, *e.g.* for 3D reconstruction. While two scenarios were taken from existing public sequences, we collected other two new outdoor scenarios and we provided an annotation for all scenarios, including those from public sources (Appendix A).

6.2 Future work

How to design compact and efficient features, which can be applied to collaborative systems with independently moving cameras while coping with severe geometric differences, is still an open challenge. Therefore, we provide future directions for our work to encourage research in this area.

Multi-scale binary descriptors discussed in this thesis, such as MORB and MST, are not as compact as single-scale descriptors, and their efficiency is highly compromised during the matching stage to find the best scales. How to provide an efficient matching strategy, which can infer the scale difference between single features in non-planar scenes, or how to design an ef-

fective reduction approach that makes the binary descriptor scale-invariant and compact are still open challenges. In Appendix C, we investigated possible scale reductions of the MST descriptor based on the principle proposed by Accumulated Stability Voting (ASV) [113]; however, the performance is still inferior with respect to MST. A possible direction could be to exploit the recent advances made in local features learned with CNN, especially at the image level as opposed to patch-based (*e.g.* see the recent GCNv2 [95] that learns binary local features to replace ORB in a SLAM framework).

The proposed cross-camera place recognition approach could become the core of a larger framework for a fully asynchronous and decentralised Collaborative Visual SLAM. While the current method addresses only the pairwise case, how to handle more than two cameras simultaneously within Collaborative SLAM requires further investigation, for example similarly to DSLAM [19]. Moreover, the framework should be systematically evaluated and compared against existing centralised and decentralised approaches [19, 74, 82, 118, 121], as the literature lacks a clear comparison and performance evaluation on datasets and scenarios recorded in situations that are more challenging than simple loop closure trajectories split into multiple sub-sequences [19], or top-down view with large overlaps between the cameras [82].

For the proposed spatio-temporal features and the cross-camera place recogntion, we assumed that the cameras are moving in a static environment, where objects cannot move and no people are present. However, most of the realistic scenarios for visual place recognition, as well as Visual SLAM, may contain motion in the scenes, which makes the view matching problem even more challenging. A common assumption is to treat features lying on moving objects as outliers and, therefore, to discard these features. It would be worth investigating how to retain these features within the framework and thus enhance the tracking of 3D objects (*e.g.* as done by CoSLAM [121] but in the more challenging case of the decentralised approach), which requires the synchronisation of the sequences as further challenge to address.

Appendix A

Dataset

We describe a set of scenarios consisting of multiple sequences that are acquired with hand-held and chest-mounted cameras (Section A.1), and we use for the evaluation of the local spatio-temporal features and the Cross-Camera Place Recognition. We use a Structure-from-Motion pipeline to obtain camera poses and calibration data, when not available a priori. While camera poses are necessary and sufficient for the proposed evaluation of the spatio-temporal features introduced in Chapter 5.3, we present here an automatic procedure that we adopt to annotate common places between sequence pairs using multi-view geometry [38] (Section A.2), as well as performance measures for evaluating XC-PR (Section A.3).

A.1 Scenarios

The dataset consists of different scenarios from publicly available datasets: TUM-RGB-D SLAM [94]; the scenario *courtyard*[1] from CoSLAM [121]; and two scenarios that we collected, *gate* and *backyard*.

From TUM-RGB-D SLAM, we use the sequences fr1_desk, fr1_desk2, and fr1_room to form the *office* scenario. Sequences were recorded at 30 fps, with a resolution of 640×480 pixels and short duration (19s, 21s, and 45s, respectively), by moving the camera slowly around a cluttered office room with different paths.

The scenario *courtyard* consists of 4 sequences (800×480 pixels), whose length varies between 3 to 4 mins, acquired around a university courtyard, starting and ending approximately

[1] drone.sjtu.edu.cn/dpzou/project/coslam.php, accessed: March 2018

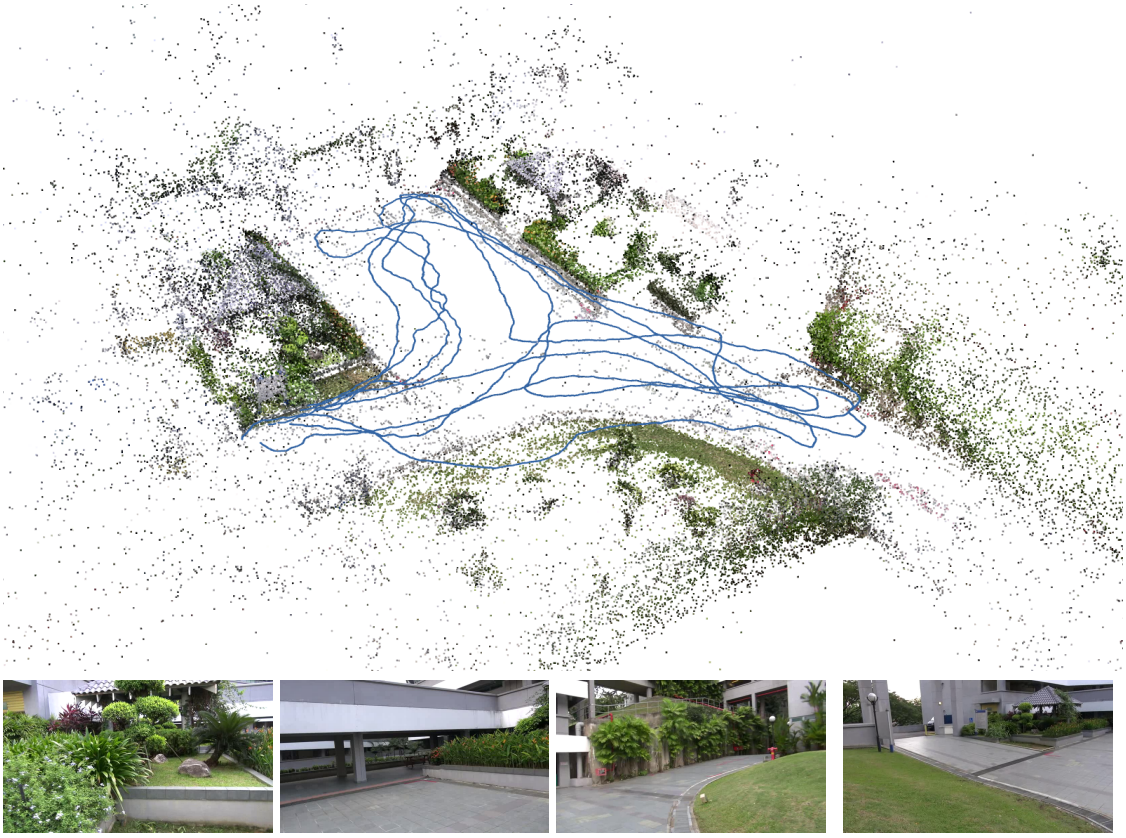


Figure A.1: Sparse reconstruction of *courtyard* using COLMAP [83]. A sample frame for each camera sequence is shown below the sparse reconstruction.

at the same positions in front of a panel, and moving the camera around the area with different paths. While the sequences were initially recorded at 50 fps, we sub-sample them to 25 fps for annotation purposes. Figure A.1 shows a sample frame for each sequence and the 3D sparse reconstruction of the scene obtained with a Structure-from-Motion pipeline, *e.g.* COLMAP [83].

Both *gate* and *backyard* consist of 4 sequences (1280×720 pixels) at 30 fps with varying duration, acquired in different outdoor scenarios with both hand-held and chest-mounted cameras. As for *courtyard*, we sub-sample *backyard* from 30 to 10 fps due to the high number of frames and for annotation purposes. All the sequences together results in a total of ∼28,000 frames and a duration of about 25 mins. Figure A.2 and Figure A.3 shows a sample frame for each sequence, and corresponding 3D reconstruction, for *gate* and *backyard*, respectively.

Table A.1 summarises the characteristics of each scenario in terms of number of frames, duration, annotation, frame-rate, resolution, environment and platform.

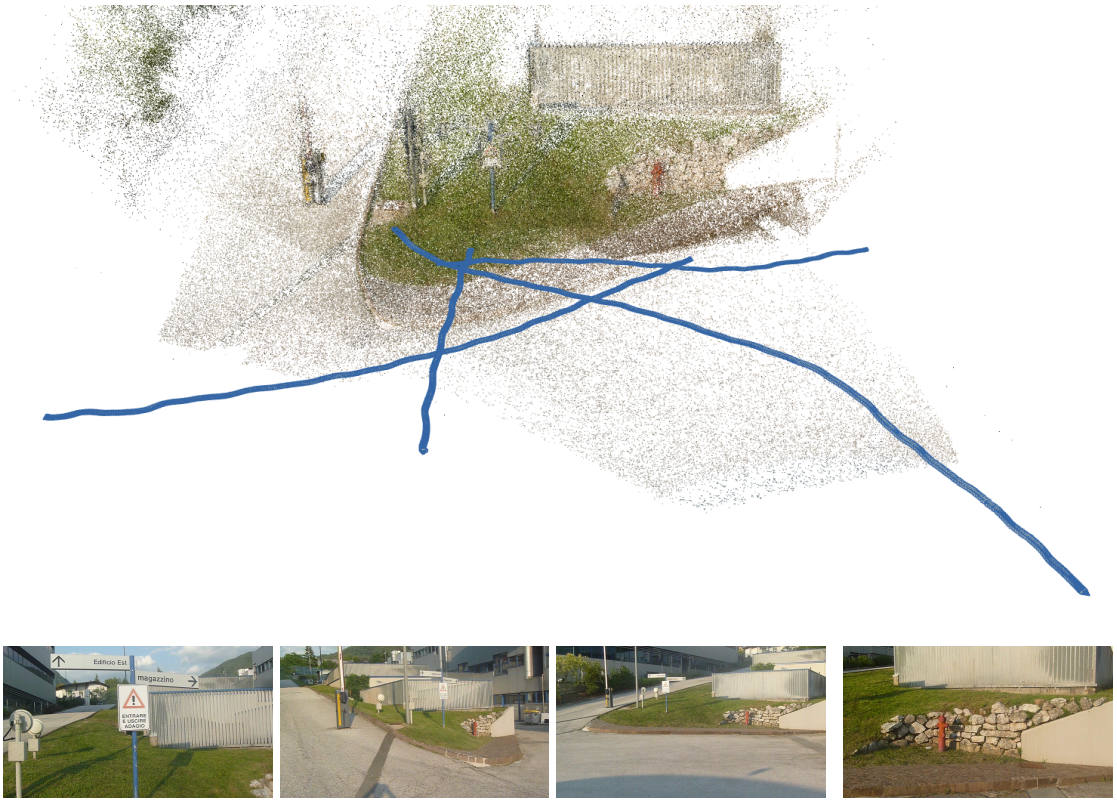


Figure A.2: Sparse reconstruction of *gate* using COLMAP [83]. A sample frame for each camera sequence is shown below the sparse reconstruction.

Table A.1: Dataset description. key – # frames: number of frames; fps: frame per second; res.: resolution; px: pixels.

Scenario	Seq.	# Frames	Duration	fps	Res. (px)	Platform
office	seq1	573	00:19	30	640x480	Hand-held
	seq2	612	00:21			
	seq3	1352	00:45			
courtyard	seq1	2849	03:10	25	800x450	Hand-held
	seq2	3118	03:28			
	seq3	3528	03:55			
	seq4	3454	03:50			
gate	seq1	330	00:11	30	1280x720	Hand-held
	seq2	450	00:15			
	seq3	480	00:16			
	seq4	375	00:13			
backyard	seq1	1217	02:02	10	1280x720	Hand-held/ Wearable
	seq2	1213	02:01			
	seq3	1233	02:03			
	seq4	1235	02:03			

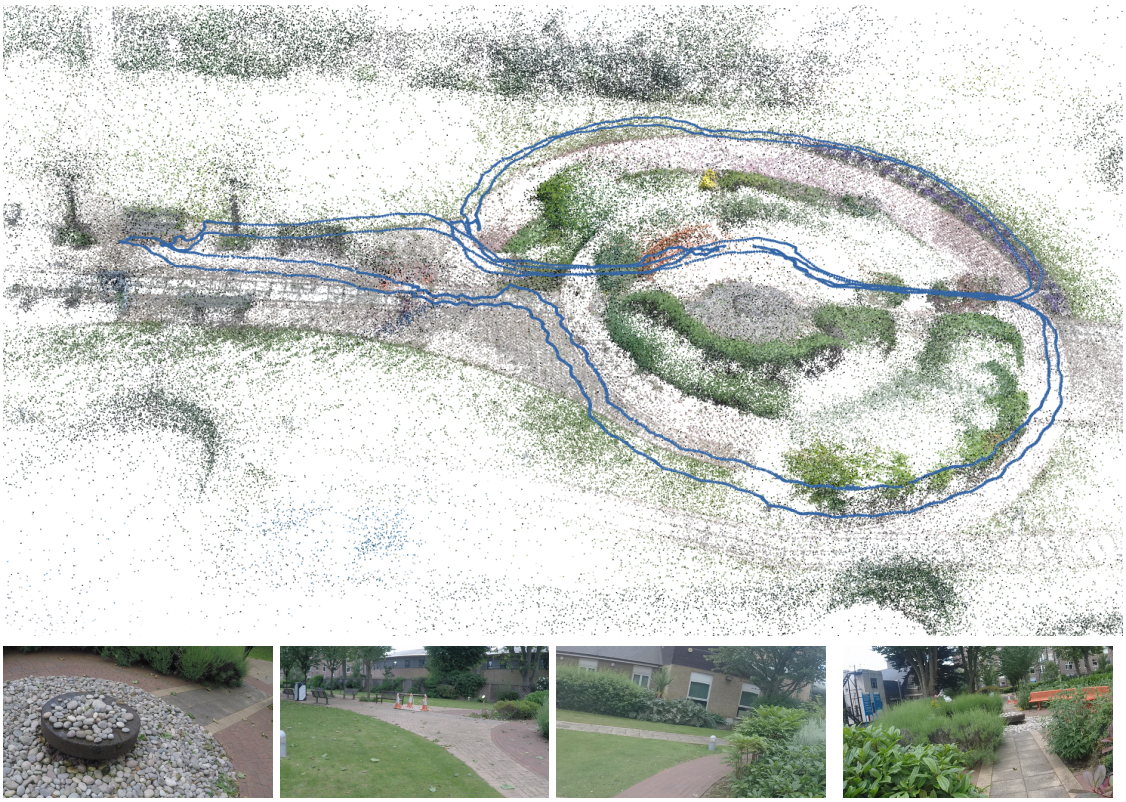


Figure A.3: Sparse reconstruction of *backyard* using COLMAP [83]. A sample frame for each camera sequence is shown below the sparse reconstruction.

A.2 Annotation

We automatically annotate views in correspondence by exploiting camera poses and calibration parameters estimated with COLMAP [83], a Structure-from-Motion pipeline, for *gate*, *courtyard*, and *backyard*, whereas these data are already available for *office*. When the frustum between two views intersects under free space assumption [66], we compute the viewpoint difference as the angular distance between the optical axes, the Euclidean distance between the camera positions, and the visual overlap as the ratio between the area spanned by projected 3D points within the image boundaries and the image area. For *office*, we localise a set of interest points (*e.g.* SIFT [55]) in the first view and we back-project the points in 3D by exploiting the depth value at the corresponding location, while we re-project the 3D points associated to the interest points of a frame in the first view onto the second view by using the estimated camera poses for the other scenarios. When annotating corresponding views, we can define a threshold on the visual overlap, *i.e.* image pairs are a valid correspondence if their visual overlap is greater than the threshold. Note that a large overlap does not imply that the viewpoint is very similar as the angular and/or Euclidean distances can be large. Moreover, we compute the total number of

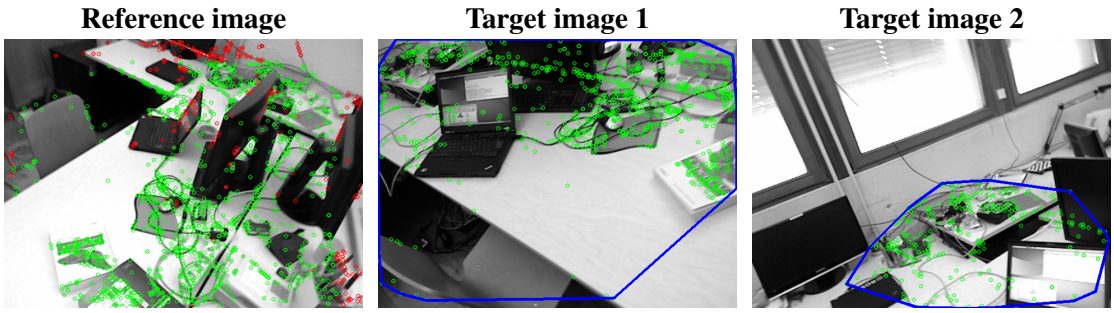


Figure A.4: Annotation of a common place (*e.g.* one side of cluttered desk) between a reference image and two target images on *office*. For each target image, we report on the bottom the overlap ratio (%), the angular distance (°) and the Euclidean distance (m) with respect to the reference image. Note the difference between the two target images. interest points detected in the reference image (red + green) are back-projected in 3D and then projected in the target image where the visual hull (—) is computed.

annotated views as the number of frames with at least one annotated view, *i.e.* a frame with more than one annotated view counts as one.

Figure A.5 shows visual overlap, angular distance and Euclidean distance between all frames in *gate 1|2*, *courtyard 1|2*, and *office 1|2*. Note that there are limited areas with highly similar viewpoints and overlap ratios. Figure A.4 shows an example of the visual hull estimated from a reference image to two target images with high ($\sim$90%) and low ($\sim$25%) overlap ratios, but large difference in the viewpoint ($> 70°$).

A.3 Performance measures

We compute place recognition accuracy and speed to evaluate the methods for XC-PR. *Place recognition accuracy* assesses the capability of a method to correctly recognise a previously seen place in a camera for each query from another camera, compared to the annotation provided with the dataset. Similarly to the evaluation of loop closure detection, we compute precision, recall, and F_1-score. *Precision* is the number of correctly recognised places over the total number of recognised places. *Recall* is the number of correctly recognised places over the total number of annotated view correspondences. *F_1-score* is the harmonic mean between precision and recall. Because of the pairwise approach, we compute precision and recall for each camera and we then average the F_1-scores between the two cameras. We use the average F_1-score to compare the methods. Moreover, we quantify the average speed of frame processing and place recognition for all methods.

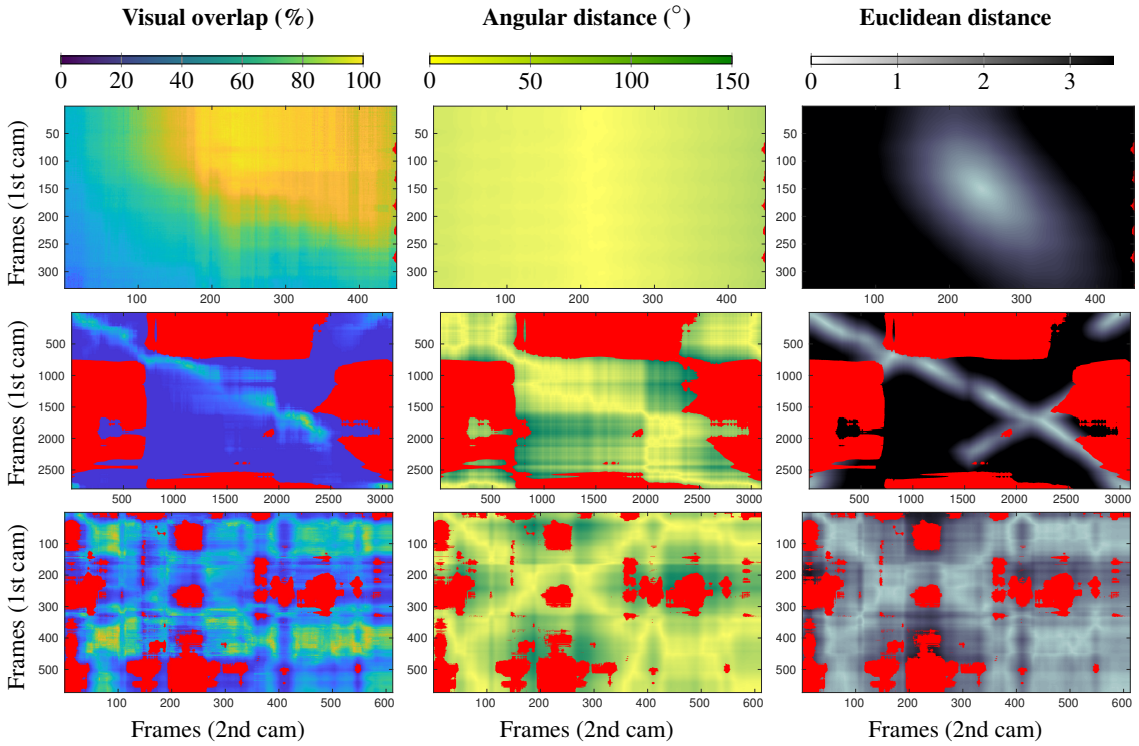


Figure A.5: Annotation heatmaps of visual overlap (ratio of the area spanned by the visual hull of re-projected 3D points and the image area) in percentage (left), angular distance between optical axes (center), and Euclidean distance between camera locations (right) in the testing sequence pairs *gate 1|2*, *courtyard 1|2*, and *office 1|2* (top-to-bottom). Note that the Euclidean distance is adimensional as the image-based 3D reconstruction is done up to an unknown scale factor, except *office*. Note also that we use different colormaps for the three measures and we denote image pairs whose frustums do not intersect with ■.

Appendix B

Scale-aware matching strategies for multi-scale binary descriptors

---

In this appendix, we discuss and analyse different scale-aware matching strategies for the multi-scale binary descriptor introduced in Chapter 3.2. We present the cross-correlation based matching strategy and variants based on varying window lengths. While these variants will take into account the detection scale, we also investigate another variant that discards the detection scale information. Experiments show that the scale-aware Hamming distance based on the *set2set mindist*, proposed in Chapter 3, leads to the best performance among these matching strategies.

B.1 Cross-correlation based distance

Let $\mathbf{d}_f$ and $\mathbf{d}_g$ be multi-scale binary descriptors of an interest point f extracted in one image and an interest point g extracted in another image, respectively. We expect the Hamming distance across scales between $\mathbf{d}_f$ and $\mathbf{d}_g$ to be similar and small for correct matches at a scale offset μ^*. Therefore, we compute as distance the cross-correlation between the two descriptors:

$$h(f,g) = \min_{\mu \in \{-(S-1),\ldots,(S-1)\}} \frac{1}{S-|\mu|} \sum_{s=|\mu+1|}^{S-|\mu|} \mathbf{d}_{f,s} \oplus \mathbf{d}_{g,s-\mu}, \tag{B.1}$$

where $\oplus$ is the XOR operator and $\mathbf{d}_{f,s} \oplus \mathbf{d}_{g,s-\mu}$ is the Hamming distance between two ORB descriptors. Figure B.1 shows an example of the cross-correlation between the descriptors extracted from the mulit-scale patches of matching points in two different images, when the scale offset is

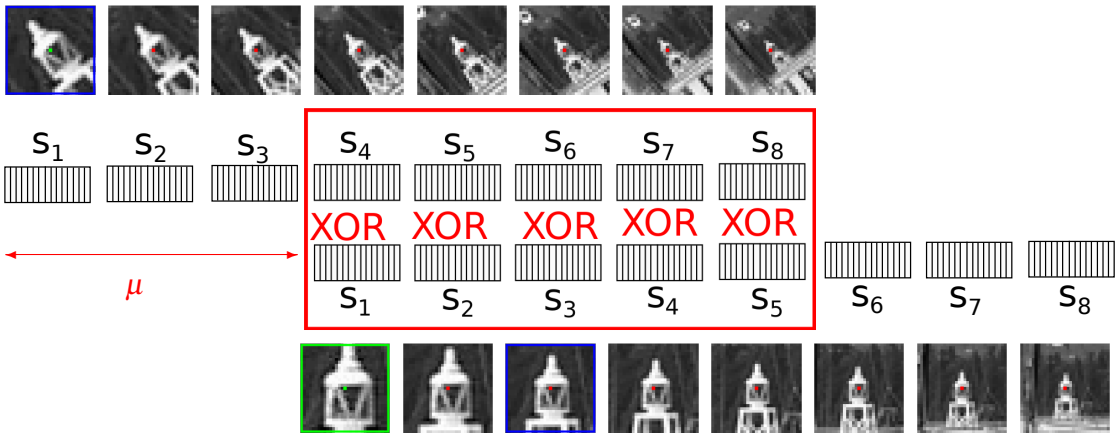


Figure B.1: Example of cross-correlation based distance estimation between two multi-scale binary descriptors using an offset $\mu = +3$. The XOR operation is computed between two single scale binary descriptors. The detection scale is denoted in blue, while the matching scale in green.

$\mu = +3$.

The formulation of Eq. B.1 discards the information about the scale where each feature was initially localised, thus leading to comparisons of descriptors at finer or coarser resolutions that might completely differ, or describe uniform areas, resulting in less distinctiveness. To take into account the detection scale, let $\hat{s}$ and $\hat{l}$ be the detection scales of feature f and g, respectively, and we re-write Eq. B.1 as follows:

$$h(i,j) = \begin{cases} \min_c \sum_{s=0}^{\Omega} \alpha_s \left[\mathbf{d}_f(\hat{l}+s+c) \oplus \mathbf{d}_g(\hat{l}+s)\right], & \text{if } c = S-\hat{l}, \\ \min_c \sum_{s=-\Omega}^{\Omega} \alpha_s \left[\mathbf{d}_f(\hat{l}+s+c) \oplus \mathbf{d}_g(\hat{l}+s)\right], & \text{if } 0 \leq c < S-\hat{l}, \\ \min_c \sum_{s=-\Omega}^{\Omega} \alpha_s \left[\mathbf{d}_f(\hat{s}+s) \oplus \mathbf{d}_g(\hat{s}+s-c)\right], & \text{if } -(S-\hat{s}) < c < 0, \\ \min_c \sum_{s=-\Omega}^{0} \alpha_s \left[\mathbf{d}_f(\hat{s}+s) \oplus \mathbf{d}_g(\hat{s}+s-c)\right], & \text{if } c = -(S-\hat{s}), \end{cases} \tag{B.2}$$

where $2\Omega+1$ is the size of the window centred at either of the two detection scales, and α_s is the weight for each scale. We introduce the weights in the formulation so that different relevance can be given finer and/or coarser scales during the alignment, however we will consider only the case of equiprobable weights in the experiments, simplifying the Eq. B.2 to a simple average instead of a weighted average. Therefore, we refer to this matching strategy based on a window cross-correlation distances as XCORR-Ω. Note that under this formulation, the matching of single descriptors extracted only at the detection scale is a special case with $\mu = \hat{s} - \hat{l}$ and $\Omega = 0$. Moreover, as by product we can compute the scale offset as $\mu^* = \hat{s} - \hat{l} + c$.

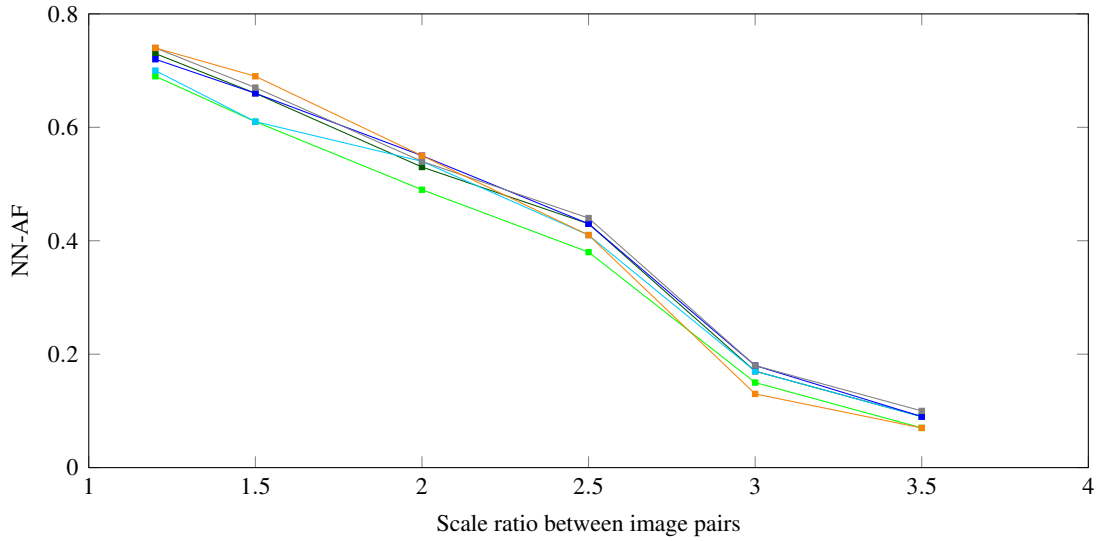


Figure B.2: Matching accuracy using the Nearest Neighbour Average F_1 score (NN-AF), when the scale ratio between image pairs increases (*i.e.* zoom in) in *venice* [42]. Legend: ■ XCORR, ■ XCORR0, ■ XCORR1, ■ XCORR2, ■ XCORR7, ■ S2SMD.

B.2 Validation

We compare the cross-correlation based matching strategies with the *set2set mindist* (S2SMD) approach used in Chapter 3. For the cross-correlation based strategies, we consider two variants: i) XCORR performs the standard correlation operation without taking into consideration the detection scale for each feature; and ii) XCORR-Ω computes the correlations with a window of size $2\Omega+1$ centred at the detection scale of each feature. We vary the window length as $\Omega = \{0,1,2,7\}$, and refer to the strategies as XCORR0, XCORR1, XCORR2, and XCORR7, respectively. For all strategies, we first localise ORB corner points [77] and we then describe the features with the MORB descriptor. We compute the cross-scale distances for each descriptor pair between two images, followed by the nearest neighbour matching strategy to find the final one-to-one matches.

Following the validation in Chapter 5, we evaluate the cross-scale matching strategies on all the image sets in the Oxford Affine Covariance Regions Dataset (ACRD) [61] and on the *venice* set from [42]. We then compute the Nearest Neighbour Average F_1 score (NN-AF) and the Matching score (MS) as performance measures. Note that for all images we localise a maximum of 1000 ORB features[1].

Figure B.2 shows that the accuracy (NN-AF) of the cross-scale matching strategies when

[1] After updating OpenCV from 3.3 to 4.1, the detected ORB features are slightly different, leading to some small variations in the reported results w.r.t. Chapter 5.

increasing the scale ratio (zoom in) between image pairs in the *venice* set [42]. Approaches follow a similar behaviour with comparable performance that considerably decreases after a scale ratio of 2.5. While S2SMD performs better than other approaches at ratios smaller than 2.5, XCORR2 and XCORR7 can slightly handle better extreme scale ratios (3 and 3.5). As it is not clear what is the impact on the performance of these matching strategies only on the *venice* set, we analyse the results on the ACRD dataset where both geometric and photometric variations are considered, as well as combination of the scale changes with in-plane rotations (*bark* and *boat* sets).

Table B.1 shows the results on the ACRD dataset. SS2MD achieves overall the best performance in terms of NN-AF and almost for each image pair in all sets. MS instead is overall similar between all the strategies. In some image pairs, XCORR and variants achieve performance comparable to or higher than S2SMD, such as the *bikes* and *ubc* sets that are affected by illumination changes and JPEG artefacts, respectively. Under in-plane rotation and scale changes (*bark* and *boat* sets), S2SMD achieves the best performance; however, under severe changes (image pair 1 – 6), the performance drastically drops down and is comparable to the performance of XCORR and variants. All strategies are not robust to viewpoint changes, as the performance considerably decreases with the increase in the transformation.

In conclusion, we observed that our initial hypothesis to compute the distance between two multi-scale binary descriptors using cross-correlation performs, overall, worse of more than 0.05 NN-AF than using a *set2set mindist* approach on different geometric and photometric transformation. Because of this observation, we chose the set2set mindist approach to validate the MORB descriptor in Chapter 5.

Table B.1: Nearest Neighbour Average F-score (NN–AF) and Matching Score (MS) for each image pair for each set of images in the ACRD dataset using different cross-scale matching strategies. For each image, a maximum number of 1000 ORB features are detected and then MORB descriptors are extracted. Best results in bold.

		NN–AF						MS					
		XCORR	XCORR-0	XCORR-1	XCORR-2	XCORR-7	S2SMD	XCORR	XCORR-0	XCORR-1	XCORR-2	XCORR-7	S2SMD
bark	1 – 2	.29	.29	.32	.30	.30	**.32**	.11	.11	.12	.11	.12	.11
	1 – 3	.13	.14	.13	.14	.14	**.15**	.04	.04	.04	.04	.04	.04
	1 – 4	.35	.30	.30	.32	.33	**.36**	.10	.08	.09	.09	.09	.10
	1 – 5	**.38**	.32	.32	.34	.35	**.38**	.10	.08	.08	.08	.08	.10
	1 – 6	**.17**	.09	.14	.16	**.17**	.10	.03	.02	.03	.03	.03	.03
bikes	1 – 2	.73	.70	.71	.71	.73	**.77**	**.61**	.53	.56	.57	**.61**	.55
	1 – 3	.68	.65	.66	.67	.69	**.73**	.55	.46	.49	.51	**.56**	.51
	1 – 4	.62	.56	.57	.59	.63	**.67**	**.45**	.36	.38	.40	**.45**	.41
	1 – 5	.54	.55	.55	.55	.55	**.57**	.36	.32	.33	.34	**.37**	.33
	1 – 6	.46	.48	.48	**.51**	.50	.48	.29	.27	.28	.30	**.31**	.26
boat	1 – 2	.58	.57	.58	.60	.59	**.66**	.49	.43	.46	.49	**.50**	.48
	1 – 3	.58	.57	.59	.59	.59	**.65**	**.44**	.38	.42	.43	**.44**	.42
	1 – 4	.44	.46	.42	.45	.44	**.51**	.29	.28	.26	.29	.29	**.30**
	1 – 5	.42	.41	.42	.42	.41	**.46**	.24	.22	.23	.24	.23	**.25**
	1 – 6	**.17**	.12	.14	.16	.16	.15	**.08**	.06	.06	.07	.07	.07
graffiti	1 – 2	.56	.56	.57	.57	.58	**.64**	.46	.40	.44	.45	**.49**	.44
	1 – 3	.26	.30	.26	.28	.28	**.34**	.21	.21	.19	.22	**.23**	**.23**
	1 – 4	.06	.11	.12	.11	.09	**.14**	.05	.07	**.09**	.08	.07	**.09**
	1 – 5	.01	.01	**.02**	**.02**	.01	.01	.00	.01	**.02**	.01	.01	.00
	1 – 6	.00	.00	**.01**	.00	.00	.00	.00	.00	.00	.00	.00	.00
leuven	1 – 2	.64	.63	.64	.63	.65	**.70**	.47	.41	.45	.43	**.48**	.45
	1 – 3	.57	.57	.58	.59	.59	**.62**	.38	.36	.36	.37	**.39**	.36
	1 – 4	.54	.52	.51	.55	.55	**.62**	.32	.29	.29	.32	.33	**.34**
	1 – 5	.47	.46	.45	.46	.50	**.56**	.26	.25	.24	.25	.28	**.29**
	1 – 6	.46	.44	.45	.45	.47	**.52**	.27	.23	.25	.25	**.28**	.27
trees	1 – 2	.48	.50	.49	.48	.51	**.59**	.32	.30	.31	.30	**.34**	**.34**
	1 – 3	.42	.45	.43	.45	.44	**.53**	.25	.24	.23	.25	**.26**	**.26**
	1 – 4	.25	.27	.27	.28	.27	**.35**	.12	.12	.12	.13	.13	**.15**
	1 – 5	.21	.28	.27	.29	.25	**.32**	.09	.11	.11	**.12**	.11	**.12**
	1 – 6	.15	.18	.18	.19	.16	**.22**	.06	.07	.07	.07	.07	**.08**
ubc	1 – 2	**.94**	.92	.93	.93	**.94**	.91	**.91**	.87	.89	.90	**.91**	.78
	1 – 3	**.90**	.89	.89	.89	**.90**	.89	**.84**	.80	.82	.83	**.84**	.74
	1 – 4	.84	.81	.82	.83	.84	**.87**	**.79**	.72	.75	.78	**.79**	.72
	1 – 5	.73	.67	.68	.70	.74	**.79**	**.69**	.56	.58	.63	**.69**	.63
	1 – 6	.61	.53	.55	.56	.63	**.67**	.53	.42	.44	.46	**.55**	.49
wall	1 – 2	.44	.54	.49	.49	.47	**.65**	.38	.39	.38	.40	.41	**.45**
	1 – 3	.34	.51	.44	.42	.38	**.62**	.29	.37	.34	.35	.32	**.44**
	1 – 4	.14	.30	.25	.25	.19	**.38**	.10	.17	.16	.16	.12	**.21**
	1 – 5	.03	.09	.08	.08	.05	**.13**	.02	.05	.04	.04	.03	**.07**
	1 – 6	.00	.01	**.01**	.00	.00	**.01**	.00	**.01**	.00	.00	.00	**.01**
ACRD avg.		.41	.42	.42	.42	.43	**.48**	.30	.28	.29	.29	**.31**	.30

Appendix C

Reducing the multi-scale temporal descriptor across scales

In this appendix, we tackle the problem to reduce the multi-scale temporal feature, MST, proposed in Chapter 3, to a more compact descriptor, which aims to be scale-invariant and whose dimensionality is represented with only 32 bytes, as if the descriptor was extracted at a single scale. Similarly to the Accumulated Stability Voting (ASV) approach [113], we propose to reduce the MST descriptor by comparing all the single binary descriptors across scale. Then, we accumulate and threshold the results to obtain a final vector of stable binary tests across scales. We applied this principle in five different variants to the MST descriptor either a frame-level or after temporal reduction. Using a scale-invariant descriptor, therefore, avoids us to compute the expensive scale-aware Hamming distance (Eq. 3.14 in Chapter 3) between feature pairs, making the matching as efficient as matching two single-scale descriptors. However, experiments show that the proposed reductions decrease the matching accuracy, hence decreasing the distinctiveness of the descriptor.

C.1 Binary accumulated stability voting

Let $\mathbf{d} \in \{0,1\}^{D \times S}$ be the D-dimensional descriptor of an interest point $\mathbf{x}$ extracted at multiple scales of an pyramid $I = \{\mathbf{I}_s\}_{s=0}^{S-1}$, where S is the number of scales. We determine the stability of each binary test across scales by computing the residual of two descriptors between any pair of

scales, (s,l), as:

$$\mathbf{r}_{s,l} = \mathbf{d}_s \oplus \mathbf{d}_l, \quad \forall s,l \in \{0,\ldots,S-1\} \wedge s \neq l. \tag{C.1}$$

The total number of estimated residuals is $R = (S-1)S/2$, as $\mathbf{r}_{s,l} = \mathbf{r}_{l,s}$. We then re-index $\mathbf{r}_{s,l}$ as $\mathbf{r}_m$, with $m = \{0,\ldots,R-1\}$ and reduce all the residuals to the fixed-length and compact descriptor $\mathbf{b} = [b_1,\ldots,b_D]$ using the following binary test for each element d:

$$b_d = \begin{cases} 1, & \text{if } \left(\sum_{m=0}^{R-1} r_{d,m}\right) < \beta \\ 0, & \text{otherwise,} \end{cases} \tag{C.2}$$

where

$$\beta = med\left(\sum_{m=0}^{R-1} r_{1,m},\ldots,\sum_{m=0}^{R-1} r_{D,m}\right), \tag{C.3}$$

is the locally determined threshold.

C.2 Discussion

The principle of this reduction is similar to ASV [113] that compares histogram-based descriptors, such as SIFT [55], at multiple scales by computing the absolute value of the difference between descriptors pairs. For each descriptor pair, then, a quantisation is performed based on a relative threshold determined by the principle of maximum entropy for each element of the resulting vector (the optimal threshold for each scale pair is given by the median of all elements [113]). The binary vectors are summed together for obtaining the final descriptor. Unlike ASV [113], the proposed reduction applies directly to the binary descriptor instead of histogram-based descriptors, and, therefore, we avoid the quantisation step before accumulating all the binary differences given by the XOR operation. ASV [113] provides a second-stage thresholding to quantise the resulting descriptor in a binary vector, using either one or multiple thresholds. We instead adopt the same principle of maximum entropy and apply the median to obtain the final binary descriptor that preserves the dimensionality of the descriptor as extracted at a single scale. Moreover, ASV-SIFT [113] was proposed for image matching, but here we consider the proposed reduction for MST features and their matching in short image sequences.

C.3 Validation

We apply the proposed reduction to MST with different five variants: i) ASV-MST direclty reduces MST across scales considering only the dominant part and discarding the stability vectors computed over time; ii) TASV-S first reduces MST for each frame after having accumulated the descriptors and then performs the temporal reduction based only on the most frequent binary tests; iii) TASV complements TASV-S with the most stable binary tests over time; iv) TASV-S* reduces online the multi-scale descriptor extracted for each frame and then tracks the reduced features before performing the temporal reduction of the most frequent binary tests; and v) complements TASV-S* with the most stable binary tests over time. We also compare the reduction variants with the baseline MST-S and MST.

We use the pairs of sequences from *office*, *desk*, *courtyard*, and *gate* as done for the validation of MST. We therefore consider precision, recall and $F1$ score as performance measures.

Table C.1 reports the matching performance of each method for each sequence, while Figure C.1 shows the F_1-score aggregated across all the sequence pairs. We can observe that overall MST-S and MST outperform the variants based on the proposed reduction in terms of F_1-score. Even though most of the reduction-based variants achieve a higher precision than MST, recall is worse as the number of retrieved matches is much lower than MST-S and MST. Among the reduction-based variants, we can observe that reducing the multi-scale temporal feature after tracking a feature point is a better choice than reducing the descriptor across scales and then tracking. Moreover, even in this experiment, it is confirmed that using the additional vector of the most stable binary tests over time slightly boost the performance (F_1-score), even if more false positives are found, for both MST, TASV, and TASV*. Finally, TASV outperforms all the reducing-based variants suggesting that performing the scale reduction before the temporal reduction is a better choice.

In conclusion, the reduction approaches allow to obtain a matching time per descriptor pair as efficient as ORB [77] features, but the performance are worse of 10% F_1-score than MST. How to determine a scale-invariant temporal descriptor for efficient binary features still remains an open challenge.

Table C.1: Matching results with the nearest neighbour strategy and Lowe's ratio test using ORB features. Best results in bold, second best in italic.

Sequence	**Method**	**Performance measures**			
		Number of matches	Precision	Recall	F_1-score
desk	MST-S	388	44.85	*13.60*	*20.88*
	MST	533	42.40	**17.67**	**24.94**
	ASV-TDS	163	54.60	6.96	12.34
	TASV-S	183	*56.28*	8.05	14.09
	TASV	292	51.03	11.65	18.97
	TASV-S*	106	**58.49**	5.17	9.49
	TASV*	166	53.61	7.42	13.03
office	MST-S	541	43.44	*8.96*	*14.85*
	MST	834	36.57	**11.63**	**17.65**
	ASV-TDS	175	48.00	3.20	6.00
	TASV-S	195	**46.67**	3.47	6.46
	TASV	419	40.33	6.44	11.11
	TASV-S*	44	*45.45*	1.18	2.30
	TASV*	97	32.99	1.89	3.57
courtyard	MST-S	1214	*85.17*	*29.99*	*44.36*
	MST	1610	74.91	**34.98**	**47.69**
	ASV-TDS	454	**86.34**	11.37	20.09
	TASV-S	491	85.13	12.12	21.22
	TASV	723	74.97	15.72	25.99
	TASV-S*	199	81.41	7.23	13.28
	TASV*	335	72.54	10.84	18.87
gate-1	MST-S	892	56.50	20.77	30.37
	MST	1293	48.18	**25.67**	**33.49**
	ASV-TDS	397	*69.77*	11.41	19.62
	TASV-S	422	68.25	11.87	20.22
	TASV	705	57.73	16.77	25.99
	TASV-S*	225	**72.89**	9.44	16.71
	TASV*	361	60.11	12.49	20.68
gate-2	MST-S	319	18.81	4.42	7.16
	MST	584	14.55	**6.26**	**8.76**
	ASV-TDS	69	**24.64**	1.25	2.38
	TASV-S	92	*22.83*	1.55	2.90
	TASV	203	18.23	2.73	4.74
	TASV-S*	56	14.29	0.68	1.30
	TASV*	131	12.98	1.45	2.61
gate-3	MST-S	1095	**55.07**	*25.36*	*34.73*
	MST	1562	46.09	**30.28**	**36.55**
	ASV-TDS	357	*71.99*	10.81	18.79
	TASV-S	425	69.65	12.45	21.12
	TASV	671	53.80	15.18	23.68
	TASV-S*	202	**73.27**	8.41	15.09
	TASV*	341	54.84	10.62	17.80

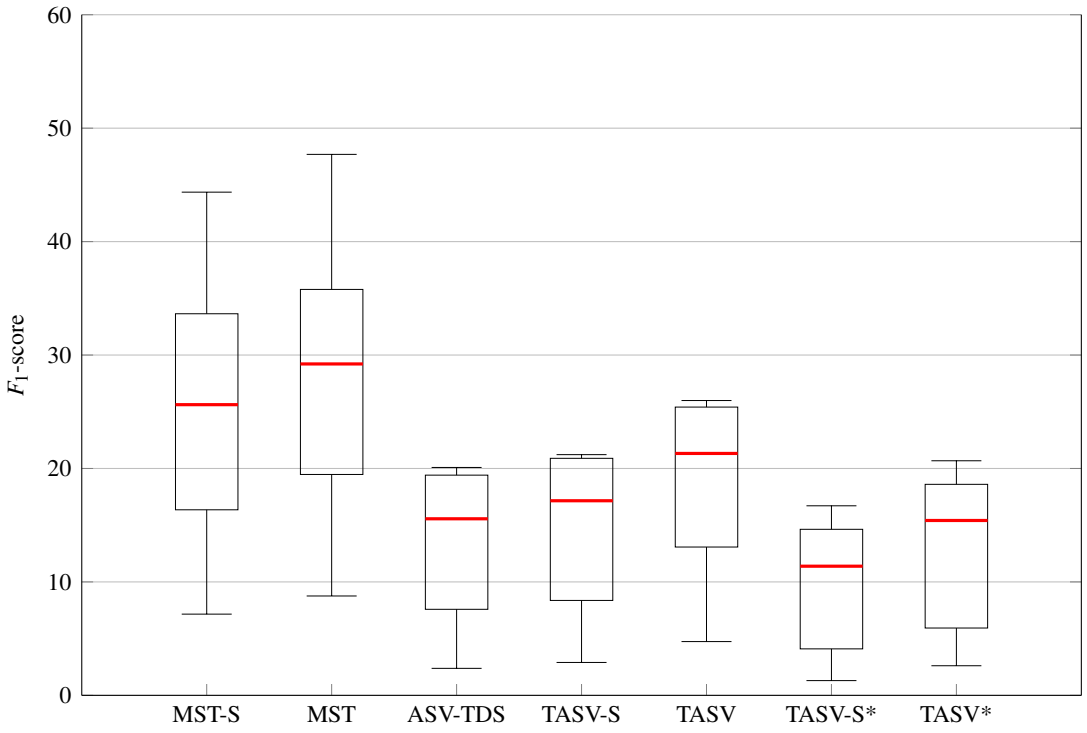


Figure C.1: Aggregated F_1-score results across all scenarios.

Bibliography

[1] H. Aanæs, A.L. Dahl, and K. Steenstrup Pedersen. Interesting Interest Points. *International Journal of Computer Vision*, 97(1):18–35, March 2012.

[2] H. Aghajan and A. Cavallaro. *Multi-Camera Networks: Principles and Applications*. Academic Press, 2009.

[3] A. Alahi, R. Ortiz, and P. Vandergheynst. FREAK: Fast retina keypoint. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, 16–21 June 2012.

[4] O. Alatas, O. Javed, and M. Shah. Compressed spatio-temporal descriptors for video matching and retrieval. In *Proceedings of the IEEE Conference on Pattern Recognition*, Cambridge, UK, 26 August 2004.

[5] P.F. Alcantarilla, J. Nuevo, and A. Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *Proceedings of the British Machine Vision Conference*, Bristol, UK, 9–13 September 2013.

[6] H. Altwaijry, A. Veit, S. J. Belongie, and C. Tech. Learning to detect and match keypoints with deep architectures. In *Proceedings of the British Machine Vision Conference*, York, UK, 19–22 September 2016.

[7] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1437–1451, June 2018.

[8] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, 16–21 June 2012.

[9] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. HPatches: a benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017.

[10] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Proceedings of the British Machine Vision Conference*, York, UK, 19–22 September 2016.

[11] V. Balntas, L. Tang, and K. Mikolajczyk. Binary Online Learned Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):555–567, March 2018.

[12] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. In *Proceedings of the European Conference on Computer Vision*, Graz, Austria, 7–13 May 2006.

[13] F. Bellavia and C. Colombo. Rethinking the sGLOH descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):931 – 944, April 2018.

[14] F. Bellavia, D. Tegolo, and C. Valenti. Keypoint descriptor matching with context-based orientation estimation. *Image and Vision Computing*, 32(9):559–567, September 2014.

[15] J. Bouguet. Pyramidal implementation of the Lucas Kanade feature tracker. Technical report, Intel Corporation, Microprocessor Research Labs, 2000.

[16] M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, 20–25 June 2005.

[17] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *Proceedings of the European Conference on Computer Vision*, Heraklion, Crete, Greece, 5–11 September 2010.

[18] F. Camposeco, A. Cohen, M. Pollefeys, and T. Sattler. Hybrid scene compression for visual localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 16–20 June 2019.

[19] T. Cieslewski, S. Choudhary, and D. Scaramuzza. Data-efficient decentralized visual SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Brisbane, Australia, 21–25 May 2018.

[20] T. Cieslewski and D. Scaramuzza. Efficient decentralized visual place recognition from full-image descriptors. In *International Symposium on Multi-Robot and Multi-Agent Systems*, Los Angeles, CA, USA, 4–5 December 2017.

[21] T. Cieslewski and D. Scaramuzza. Efficient decentralized visual place recognition using

a distributed inverted index. *IEEE Robotics and Automation Letters*, 2(2):640–647, April 2017.

[22] G. Csurka and M. Humenberger. From handcrafted to deep local invariant features. arXiv:1807.10254v3 [cs.CV], June 2019.

[23] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, 20–25 June 2005.

[24] D. DeTone, T. Malisiewicz, and A. Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Salt Lake City, UT, USA, 18–22 June 2018.

[25] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *The IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Beijing, China, 15–16 October 2005.

[26] J. Dong and S. Soatto. Domain-size pooling in local descriptors: DSP-SIFT. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 7–12 June 2015.

[27] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-Net: A trainable cnn for joint description and detection of local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 16–20 June 2019.

[28] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625, March 2018.

[29] B. Fan, Q. Kong, T. Trzcinski, Z. Wang, C. Pan, and P. Fua. Receptive fields selection for binary feature description. *IEEE Transactions on Image Processing*, 23(6):2583–2595, June 2014.

[30] P. Fischer, A. Dosovitskiy, and T. Brox. Descriptor matching with convolutional neural networks: a comparison to SIFT. arXiv:11405.5769v2 [cs.CV], June 2015.

[31] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[32] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza. Collaborative monocular SLAM with multiple micro aerial vehicles. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems*, Tokyo, Japan, 3–7 November 2013.

[33] D. Gálvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, October 2012.

[34] X. Gao, R. Wang, N. Demmel, and D. Cremers. LDSO: Direct sparse odometry with loop closure. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems*, Madrid, Spain, 1–5 October 2018.

[35] Z. Gao, W. Nie, A. Liu, and H. Zhang. Evaluation of local spatial–temporal features for cross-view action recognition. *Neurocomputing*, 173(P1):110–117, January 2016.

[36] E. Garcia-Fidalgo and A. Ortiz. iBoW-LCD: An appearance-based loop-closure detection approach using incremental bags of binary words. *IEEE Robotics and Automation Letters*, 3(4):3051–3057, October 2018.

[37] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the fourth Alvey Vision Conference*, Manchester, UK, 31 August–2 September 1988.

[38] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Second edition, 2003.

[39] T. Hassner, S. Filosof, V. Mayzels, and L. Zelnik-Manor. SIFTing through scales. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1431–1443, July 2017.

[40] T. Hassner, V. Mayzels, and L. Zelnik-Manor. On SIFTs and their scales. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, 16–21 June 2012.

[41] K. He, Y. Lu, and S. Sclaroff. Local descriptors optimized for average precision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18–22 June 2018.

[42] J. Heinly, E. Dunn, and J. Frahm. Comparative evaluation of binary features. In *Proceedings of the European Conference on Computer Vision*, Firenze, Italy, 7–13 October 2012.

[43] J. Heinly, J. L. Schönberger, E. Dunn, and J. Frahm. Reconstructing the world* in six days *(as captured by the Yahoo 100 million image dataset). In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.

[44] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, 27 June–2 July 2004.

[45] A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *Proceedings of the British Machine Vision Conference*, Leeds, UK, 1–4 September 2008.

[46] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality*, Nara, Japan, 13–16 November 2007.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances on Neural Information Processing and Systems*, Lake Tahoe, NV, USA, 2012.

[48] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, September 2005.

[49] I. Laptev, C. Marszalek, M. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 23–28 June 2008.

[50] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *Proceedings of the IEEE International Conference on Computer Vision*, Barcelona, Spain, 6–13 November 2011.

[51] G. Levi and T. Hassner. LATCH: learned arrangements of three patch codes. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, Lake Placid, NY, USA, 7–9 March 2016.

[52] R. Leyva, V. Sanchez, and C. Li. Compact and low-complexity binary feature descriptor and Fisher vectors for video analytics. *IEEE Transactions on Image Processing*, 28(12):6169–6184, Dec 2019.

[53] K. Lin, J. Lu, C. Chen, J. Zhou, and M. Sun. Unsupervised deep learning of compact binary

descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(6):1501–1514, June 2019.

[54] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the IEEE International Conference on Computer Vision*, Kerkyra, Greece, 20–27 September 1999.

[55] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[56] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, February 2016.

[57] Z. Luo, T. Shen, L. Zhou, S. Zhu, R. Zhang, Y. Yao, T. Fang, and L. Quan. GeoDesc: Learning Local Descriptors by Integrating Geometry Constraints. In *Proceedings of the European Conference on Computer Vision*, Munich, Germany, 14–18 September 2018.

[58] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Verlag, 2005.

[59] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *Proceedings of the European Conference on Computer Vision*, Heraklion, Crete, Greece, 5–11 September 2010.

[60] K. Mikolajczyk and C. Schmid. Scale & Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60(1):63–86, October 2004.

[61] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, October 2005.

[62] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances on Neural Information Processing and Systems*, Long Beach, CA, USA, 4–9 December 2017.

[63] R. Mitra, N. Doiphode, U. Gautam, S. Narayan, S. Ahmed, S. Chandran, and A. Jain. A large dataset for improving patch matching. arXiv:1801.01466v3 [cs.CV], April 2018.

[64] R. Mitra, J. Zhang, S. Narayan, S. Ahmed, S. Chandran, and A. Jain. Improved descriptors for patch matching and reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, Venice, Italy, 22–29 October 2017.

[65] K. Moo Yi, Y. Verdie, P. Fua, and V. Lepetit. Learning to assign orientations to feature points. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 27–30 June 2016.

[66] P. Moulon, P. Monasse, R. Perrot, and R. Marlet. OpenMVG: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, 2016.

[67] M. Muja and D. G. Lowe. Fast matching of binary features. In *Proceedings of the Conference on Computer and Robot Vision*, Toronto, Ontario, Canada, 28–30 May 2012.

[68] R. Mur-Artal, J.M.M. Montiel, and J.D. Tardos. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, October 2015.

[69] R. Mur-Artal and J. D. Tardós. Fast relocalisation and loop closing in keyframe-based SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Hong Kong, China, 31 May–5 June 2014.

[70] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, 17–22 June 2006.

[71] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, May 2001.

[72] Y. Ono, E. Trulls, P. Fua, and K. M. Yi. LF-Net: Learning local features from images. In *Advances on Neural Information Processing and Systems*, Montréal, Canada, 3–8 December 2018.

[73] J. Revaud, F. Weinzaepfel, C. Roberto de Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger. R2D2: repeatable and reliable detector and descriptor. In *Advances on Neural Information Processing and Systems*, Vancouver, Canada, 8–14 December 2019.

[74] L. Riazuelo, J. Civera, and J. M. M. Montiel. C^2TAM: A Cloud framework for Cooperative Tracking and Mapping. *Robotics and Autonomous Systems*, 62(4):401 – 413, April 2014.

[75] P. L. Rosin. Measuring corner properties. *Computer Vision and Image Understanding*, 73(2):291–307, February 1999.

[76] E. Rosten and T. Drummond. Machine Learning for High-Speed Corner Detection. In *Proceedings of the European Conference on Computer Vision*, Graz, Austria, 7–13 May 2006.

[77] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision*, Barcelona, Spain, 6–13 November 2011.

[78] J. C. SanMiguel, C. Micheloni, K. Shoop, G. L. Foresti, and A. Cavallaro. Self-reconfigurable smart camera networks. *Computer*, 47(5):67–73, May 2014.

[79] D. Schlegel and G. Grisetti. HBST: A Hamming distance embedding binary search tree for feature-based visual place recognition. *IEEE Robotics and Automation Letters*, 3(4):3741–3748, October 2018.

[80] T. Schmidt, R. Newcombe, and D. Fox. Self-supervised visual descriptor learning for dense correspondence. *IEEE Robotics and Automation Letters*, 2(2):420–427, April 2017.

[81] P. Schmuck and M. Chli. Multi-UAV collaborative monocular SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Singapore, 29 May–3 June 2017.

[82] P. Schmuck and M. Chli. CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams. *Journal of Field Robotics*, 36(4):763–781, June 2019.

[83] J. L. Schönberger and J. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 27–30 June 2016.

[84] J. L. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017.

[85] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional SIFT descriptor and its application to action recognition. In *Proceedings of the ACM International Conference on Multimedia*, Augsburg, Germany, 25–29 September 2007.

[86] X. Shen, C. Wang, X. Li, Z. Yu, J. Li, C. Wen, M. Cheng, and Z. He. RF-Net: An end-to-end image matching network based on receptive field. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 16–20 June 2019.

[87] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 21–23 June 1994.

[88] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 7–13 December 2015.

[89] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, Banff, Canada, 14–16 April 2014.

[90] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, Nice, France, 11–17 October 2003.

[91] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. In *ACM SIGGRAPH 2006 Papers*, Boston, USA, 30 July–3 August 2006.

[92] C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua. LDAHash: Improved matching with smaller descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):66–78, January 2012.

[93] C. Strecha, W. Von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 23–28 June 2008.

[94] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems*, Algarve, Portugal, 7–12 October 2012.

[95] J. Tang, L. Ericson, J. Folkesson, and P. Jensfelt. GCNv2: Efficient correspondence prediction for real-time SLAM. *IEEE Robotics and Automation Letters*, 4(4):3505–3512, October 2019.

[96] Y. Tian, B. Fan, and F. Wu. L2-Net: Deep learning of discriminative patch descriptor in Euclidean distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017.

[97] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas. SOSNet: Second order similarity regularization for local descriptor learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 16–20 June 2019.

[98] E. Tola, V. Lepetit, and P. Fua. DAISY: An efficient dense descriptor applied to wide baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830, May 2010.

[99] E. Trulls, A. Sanfeliu, and F. Moreno-Noguer. Spatiotemporal descriptor for wide-baseline stereo reconstruction of non-rigid and ambiguous scenes. In *Proceedings of the European Conference on Computer Vision*, Firenze, Italy, 7–13 October 2012.

[100] T. Trzcinski, C. M. Christoudias, and V. Lepetit. Learning image descriptors with boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):597–610, March 2015.

[101] T. Trzcinski and V. Lepetit. Efficient discriminative projections for compact binary descriptors. In *Proceedings of the European Conference on Computer Vision*, Firenze, Italy, 7–13 October 2012.

[102] K. A. Tsintotas, L. Bampis, and A. Gasteratos. Probabilistic appearance-based place recognition through bag of tracked words. *IEEE Robotics and Automation Letters*, 4(2):1737–1744, April 2019.

[103] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, June 2008.

[104] Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit. TILDE: A Temporally Invariant Learned DEtector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 7–12 June 2015.

[105] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *Proceedings of the British Machine Vision Conference*, London, UK, 7–10 September 2009.

[106] Z. Wang, B. Fan, G. Wang, and F. Wu. Exploring local and overall ordinal information for robust feature description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2198–2211, November 2016.

[107] Z. Wang, B. Fan, and F. Wu. Local intensity order pattern for feature description. In *Proceedings of the IEEE International Conference on Computer Vision*, Barcelona, Spain, 6–13 November 2011.

[108] F. Warburg, S. Hauberg, M. Lopez-Antequera, P. Gargallo, Y. Kuang, and J. Civera. Map-

illary street-level sequences: A dataset for lifelong place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 16–18 June 2020.

[109] G. Willems, T. Tuytelaars, and L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proceedings of the European Conference on Computer Vision*, Marseille, France, 12–18 October 2008.

[110] R. Williams, B. Konev, and F. Coenen. Scalable distributed collaborative tracking and mapping with micro aerial vehicles. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems*, Hamburg, Germany, 28 September–2 October 2015.

[111] S. Winder and M. Brown. Learning local image descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, USA, 17–22 June 2007.

[112] C. Wu. SiftGPU: A GPU implementation of scale invariant feature transform (SIFT). `http://cs.unc.edu/~ccwu/siftgpu`, 2011.

[113] T. Yang, Y. Lin, and Y. Chuang. Accumulated Stability Voting: A robust descriptor from descriptors of multiple scales. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 27–30 June 2016.

[114] X. Yang and K. T. Cheng. Local difference binary for ultrafast and distinctive feature description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):188–194, January 2014.

[115] J. Ye, S. Zhang, T. Huang, and Y. Rui. CDbin: Compact discriminative binary descriptor learned with efficient neural network. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–13, January 2019.

[116] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned Invariant Feature Transform. In *Proceedings of the European Conference on Computer Vision*, Amsterdam, The Netherlands, 8–16 October 2016.

[117] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 7–12 June 2015.

[118] H. Zhang, X. Chen, H. Lu, and J. Xiao. Distributed and collaborative monocular si-

multaneous localization and mapping for multi-robot systems in large-scale environments. *International Journal of Advanced Robotic Systems*, 15(3):1–20, May 2018.

[119] Z. Zhang. A flexible new technique for camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, November 2000.

[120] C. L. Zitnick and K. Ramnath. Edge foci interest points. In *Proceedings of the IEEE International Conference on Computer Vision*, Barcelona, Spain, 6–13 November 2011.

[121] D. Zou and P. Tan. CoSLAM: Collaborative visual SLAM in dynamic environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):354–366, February 2013.